Depth-based Multi-View 3D Video Coding

Zamarin, Marco

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Depth-based Multi-View 3D Video Coding

Marco Zamarin

June 2013

DTU Fotonik
Department of Photonics Engineering

Coding & Visual Communication
DTU Fotonik
Technical University of Denmark
2800 Kgs. Lyngby
DENMARK

To my family

Abstract

The interest in Three-Dimensional Video (3DV) technologies has grown considerably in both the academic and industrial worlds in the recent years. A simple and flexible 3DV representation is the so-called Multi-view Video-plus-Depth (MVD), in which depth maps are provided together with multi-view video. Depth maps are typically used to synthesize the desired output views, and the performance of view synthesis algorithms strongly depends on the accuracy of depth information.

In this thesis, novel algorithms for efficient depth map compression in MVD scenarios are proposed, with particular focus on edge-preserving solutions. In a proposed scheme, texture-depth correlation is exploited to predict surface shapes in the depth signal. In this way depth coding performance can be improved in terms of both compression gain and edge-preservation. Another solution proposes a new intra coding mode targeted to depth blocks featuring arbitrarily-shaped edges. Edge information is encoded exploiting previously coded edge blocks. Integrated in H.264/AVC, the proposed mode allows significant bit rate savings compared with a number of state-of-the-art depth codecs. View synthesis performances are also improved, both in terms of objective and visual evaluations. Depth coding based on standard H.264/AVC is explored for multi-view plus depth image coding. A single depth map is used to disparity-compensate multiple views and allow more efficient coding than H.264 MVC at low bit rates. Lossless coding of depth maps is also addressed: an efficient scheme for stereoscopic disparity maps based on bit-plane decomposition and context-based arithmetic coding is proposed. Inter-view redundancy is exploited by means of disparity warping. Major gains in compression efficiency are noticed when comparing with a number of standard solutions for lossless coding.

New approaches for distributed video-plus-depth coding are also presented in this thesis. Motion correlation between the two signals is exploited at the decoder side to improve the performance of the side information generation algorithm. In addition, optical flow techniques are used to extract dense motion information and generate improved candidate side information. Multiple candidates are merged employing multi-hypothesis strategies. Promising rate-distortion performance improvements compared with state-of-the-art Wyner-Ziv decoders are reported, both when texture is exploited to encode depth maps, and vice versa.

Resumé

Interessen for tre-dimensionale video (3DV) teknologier er vokset betydeligt i både den akademiske og industrielle verden i de seneste år. En simpel og fleksibel 3DV repræsentation er den såkaldte multi-view video-plus-depth (MVD), hvor dybden (depth) er angivet sammen med multi-view video. Depth maps bruges typisk til at syntetisere det ønskede output view og kvaliteten af view syntese algoritmer afhænger i høj grad af præcisionen af dybde informationen.

I denne afhandling foreslås nye algoritmer til effektiv depth map kompression i MVD scenarier, med særlig fokus på kant-bevarende løsninger. I en ny metode bliver texture-depth korrelation udnyttet til at forudsige overfladeformer i depth signalet. På denne måde kan depth kodning ydelsen forbedres både med hensyn til kompression og kant-bevarelse. En anden løsning indfører en ny intra kodningstilstand målrettet mod depth blokke med arbitrært-formede kanter. Kant informationen er kodet ved brug af tidligere kodede kantblokke. Den foreslåede mode integreret i H.264/AVC giver betydelig bitbesparelse sammenlignet med en række state-of-the-art dybde codecs. Syntesen er også forbedret, både med hensyn til objektive og visuelle vurderinger. Depth kodning baseret på standard H.264/AVC er undersøgt for multi-view plus depth billedkodning. En enkelt depth map bruges til at disparity-kompensere flere views og give mere effektiv kodning end H.264 MVC ved lave bithastigheder. Tabsfri kodning af depth maps behandles også: en effektiv ordning for stereoskopiske disparity maps baseret på bit-plane dekomposition og kontekst-baseret aritmetisk kodning foreslås. Inter-view redundans udnyttes ved hjælp af disparity warping. Større gevinster i kompressionseffektivitet opnås, når man sammenligner med en række standard løsninger til tabsfri kodning.

Nye metoder til distribueret video-plus-depth kodning er også præsenteret i denne afhandling. Bevægelses-korrelation mellem de to signaler udnyttes på dekodersiden til at forbedre genereringen af side information. Desuden bruges optisk-flow teknikker til at udregne motion information med høj tæthed og generere forbedret kandidat side information. Flere kandidater er lagt sammen ved at benytte multi-hypotese strategier. Lovende rate-forvrængning forbedringer sammenlignet med state-of-the-art Wyner-Ziv dekodere er opnået, både når tekstur udnyttes til at indkode depth, og omvendt.

Acknowledgements

Three years of study, research, work – but also traveling, networking, and social activities – have passed. It has been a long journey: a journey across experiences and people that I will never forget. Not everyone might be listed here, but all of you are in my memory.

I would like to thank my supervisor Prof. Søren Forchhammer for the possibility of studying in his group as a Ph.D. student, the freedom to focus my research on the topics I found more interesting and promising, and his precious comments and suggestions that in some cases really made the difference. I would also like to thank Prof. Antonio Ortega for his careful support and supervision during my stay at the University of Southern California. His genuine passion for the research and his ability to go in deep while keeping things simple and clear make him an example to follow.

I would like to thank the co-authors of my papers, especially Matteo Salmistraro, who never got tired of offering his help whenever I needed, and showed me how with perseverance and a PC any problem can be solved. Thanks to Pietro Zanuttigh, Simone Milani, and Guido M. Cortelazzo for their collaboration during these three years. Thanks to colleagues in the group and department for their support, in particular Nino Burini, who together with Federica Genovese has always been available to discuss any matters – inside and outside the university – and spend good time together.

Thanks to the Otto Mønsteds Fond and Oticon Fonden for supporting my research activities in multiple occasions.

Thanks to my Italian friends for their remote support. Special thanks to Paolo Crivellari and Lucia Lazzaretto, who even drove to Denmark to show once more how much they care. Thanks to Marco Levorato,

Tolian Gjika, Ibrahim Khalife, Aldo Della Ragione, David Jacobs, Jim Tortorelli, my cousin Nicolino, and the other guys who made my stay in California comfortable and unforgettable. You are the best memories of that period I have.

Very special thanks to Anna Ukhanova for her endless support and attention. Never too busy nor tired to assist me, she always did her best to make sure I was happy with my work, bringing lots of energy and motivation on the way. Sincere thanks for your priceless care.

The biggest thank goes to Him who made all this possible and my entire fantastic family for their constant presence, support and care in these three long years. Always ready to do everything in your power to help me, you have been and are an unmovable reference point.

Marco Zamarin

Ph.D. Publications

This thesis is based on the following original peer-reviewed publications:

PAPER 1 **M. Zamarin**, S. Forchhammer, "Edge-preserving Intra Mode for Efficient Depth Map Coding based on H.264/AVC", *Elsevier Signal Processing: Image Communication (submitted).*

PAPER 2 M. Salmistraro, L.L. Rakêt, **M. Zamarin**, A. Ukhanova, S. Forchhammer, "Texture Side Information Generation for Distributed Coding of Video-Plus-Depth", *2013 IEEE Int'l Conf. on Image Processing (ICIP 2013)*, Melbourne, Australia, Sep. 15-18, 2013 *(accepted).*

PAPER 3 **M. Zamarin**, M. Salmistraro, S. Forchhammer, A. Ortega, "Edge-preserving Intra Depth Coding based on Context-coding and H.264/AVC", *2013 IEEE Int'l Conf. on Multimedia and Expo (ICME 2013)*, San Jose, CA, USA, July 15-19, 2013 *(accepted).*

PAPER 4 M. Salmistraro, **M. Zamarin**, L.L. Rakêt, S. Forchhammer, "Distributed Multi-Hypothesis Coding of Depth Maps using Texture Motion Information and Optical Flow", *Proc. of 2013 IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, pp. 1685-1689, Vancouver, Canada, May 26-31, 2013.

PAPER 5 M. Salmistraro, **M. Zamarin**, S. Forchhammer, "Wyner-Ziv Coding of Depth Maps Exploiting Color Motion Information", *Proc. of 2013 IS&T/SPIE Visual Information Processing and Communication Conf. (EI 2013)*, Burlingame, CA, USA, Feb. 3-7, 2013.

PAPER 6 M. Zamarin, S. Forchhammer, "Lossless Compression of Stereo Disparity Maps for 3D", *Proc. of 2012 IEEE Int'l Work. on Hot Topics in 3D Multimedia (Hot3D 2012)*, pp. 617-622, Melbourne, Australia, July 13, 2012.

PAPER 7 S. Milani, P. Zanuttigh, **M. Zamarin**, S. Forchhammer, "Efficient Depth Map Compression Exploiting Segmented Color Data", *Proc. of 2011 IEEE Int'l Conf. on Multimedia and Expo (ICME 2011)*, pp. 1-6, Barcelona, Spain, July 11-15, 2011.

PAPER 8 M. Zamarin, S. Milani, P. Zanuttigh, G.M. Cortelazzo, S. Forchhammer, "A Joint Multi-View Plus Depth Image Coding Scheme Based on 3D-Warping", *Proc. of 1st ACM Int'l Work. on 3D Video Processing (3DVP 2010)*, pp. 7-12, Florence, Italy, Oct. 29, 2010.

Other publications produced during the Ph.D. not included in this thesis:

[P9] M. Salmistraro, **M. Zamarin**, S. Forchhammer, "Multi-hypothesis Distributed Stereo Video Coding", *2013 IEEE Int'l Work. on Multimedia Signal Processing (MMSP 2013)*, Pula, Italy, Sep. 30 - Oct. 2, 2013 *(accepted)*.

[P10] **M. Zamarin**, S. Forchhammer, "Coding of Depth Images for 3DTV," *3rd Annual Danish-Californian Work. on Photonic Technologies and Applications*, Berkeley, CA, USA, Jan. 30-31, 2012.

[P11] **M. Zamarin**, S. Forchhammer, "3D Video Compression and Transmission," *2nd Annual Danish-Californian Work. on Photonic Technologies for Access and Biophotonics*, Stanford, CA, USA, Jan. 31 - Feb. 1, 2011.

[P12] S. Forchhammer, M. Danieli, N. Burini, **M. Zamarin**, A. Ukhanova, "Maximizing Entropy of Image Models for 2-D Constrained Coding," *Proc. of 2010 Work. on Information Theoretic Methods in Science and Engineering (WITMSE 2010)*, Tampere, Finland, Aug. 16-18, 2010.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Similarly to the transition from black and white television to color television, Three-Dimensional Video (3DV) is believed to be the next major development step in the evolution of motion picture formats towards a more natural and realistic visual experience. However, Three-Dimensional Television (3DTV) is not a new concept at all. The invention of the first system enabling stereoscopic vision of still images dates back to 1838, when a mirror-based device able to fuse two different perspective images into one and give the viewer the impression of depth was invented [1]. Later in the 1920's the first full-length 3D film was released and an experimental stereoscopic TV setup was demonstrated, thus showing that 3DTV is almost as old as its monoscopic counterpart – also demonstrated in the 1920's. The interest in 3D movies had the first peak in the 1950's, when the cinema industry tried to react to the increasing popularity of television. However, due to the lack of experience with the new technology and the big limitations of the display equipments, the production of 3D movies significantly dropped in the late 1950's without showing a clear return before the 2000's [2]. The interest in 3DTV followed the same trend until 1996, when the International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) Joint Technical Committee (JTC) 1 / Sub-Committee (SC) 29 / Working Group (WG) 11 – Moving Pictures Experts Group (MPEG) defined the MPEG-2 Multiview Profile [3] as

an amendment to the H.262/MPEG-2 Video standard [4] targeted to the encoding of stereoscopic and multi-view video. Despite the standardization efforts, the multi-view extension was never implemented in actual products as the market – in the middle of the transition between analog and digital video services – was not ready to support such a novel technology.

Later in 2008 the successful H.264/MPEG-4 Advanced Video Coding (AVC) [5] standard was extended with the Multi-view Video Coding (MVC) [6] extension featuring more efficient inter-view prediction schemes and higher compression gains. However, it was clear that in order to support the large number of views required by high-quality autostereoscopic multi-view displays new solutions were needed: the size of an MVC bit stream is in fact proportional to the number of views in the multi-view video [7]. Motivated by this issue and other limitations of both MPEG-2 Multiview Profile and MVC, a new approach to 3DTV that efficiently exploits the scene geometry was proposed [8]. After a first exploration phase, the technology is being now standardized by the MPEG 3DV ad-hoc group [9], which recently joined the efforts with the International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) Study Group (SG) 16 Working Party (WP) 3 to form the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) [10]. The idea is to represent a 3DV with a small number of views – typically 2 to 3 – and the corresponding depth map information [11]. An example of a view and the associated depth map is provided in Fig. 1.1. If depth data are available at the decoder side, *any* intermediate view between the 2 or 3 input views can be synthesized by means of Depth-Image-Based-Rendering (DIBR) algorithms [12] according to the specific display requirements, and used for display. This format looks particularly attractive as it allows decoupling content creation and display requirements – a key feature for the success of 3DTV delivery [13] – without introducing substantial bit rate increases, thus solving the central issue of MVC. Both stereoscopic and multi-view displays can take advantage of this format: the former to adapt the content to the specific stereo baseline and allow users to adjust the depth perception, the latter to render a large number of output views and cover wide viewing angles [14]. Depth images not only enable the generation of virtual views for stereoscopic vision: due to the fact that any intermediate

(a) (b)

Figure 1.1: A color image (a) and the corresponding depth map (b) from the *Ballet* dataset [17]. The brighter the pixels, the closer the object. Frames of other test sequences are reported in App. B.

view can be generated at the decoder side, they allow convenient implementations of the so-called Free Viewpoint Video (FVV) technology, in which the user can freely navigate – i.e. change the viewpoint – around the scene within the limits of the input views [15, 16]. Stereoscopic vision and FVV can be also combined for highly immersive viewing experiences.

Depth maps consist of gray-scale images mapping distances between objects in the scene and the camera plane of the physical/virtual depth camera to gray intensities, pixel per pixel. Efficient encoding of depth information is of crucial importance in DIBR-based 3DV communication systems as the accuracy of depth information strongly affects the performance of view synthesis algorithms [18]. Depth maps can be compressed with standard video codecs, but the substantial differences between natural images and depth information make this approach far from optimal. Depth maps are characterized by smooth regions divided by sharp edges, which play an important role in the view synthesis process. More than 200 schemes have been proposed in the last years in the literature[1] for efficient coding of depth data, with particular focus on edge preservation (see Sec. 1.2.4). However, the problem of efficient edge-aware depth coding remains challenging: an ultimate solution has not been found yet and the possibilities of improvement are multiple. Joint texture (color) and depth coding exploiting shape and motion correlations between the

[1] Based on a search in the IEEE and SPIE Digital Libraries in March 2013.

two signals has also been explored, showing that some additional gains in terms of compression efficiency are possible with different setups when the two components are jointly encoded.

1.2 Background

In this Section background information on representations of 3D scenes, rendering of virtual views, acquisition and estimation of depth information, compression and transmission of 3DV are provided. The Section concludes with some research trends in the field of 3DV representation and transmission exploiting geometry information. For more information the reader is referred to [19–22].

1.2.1 3D Scene Representation and View Rendering

Many diverse representations of 3D scenes have been proposed in the literature. Each one presents different advantages and disadvantages, so the choice of one over the others depends on the requirements of the specific 3DV system. In general, 3D scene representation methods can be classified in relation to camera density, as proposed in the computer graphics literature [23, 24]. Two extremes are identified: image-based representations and geometry-based representations. Following this classification, Fig. 1.2 shows an overview of the most common representations. On one hand image-based representations (including Ray-Space and light-field) typically require very dense camera settings in order to ensure good rendering performance. View generation is performed by means of interpolation from the available camera views without basing on any geometric model. In this approach complexity is a major issue as the amount of data to be processed is huge. On the other hand geometry-based representations may require less dense camera settings but rely on complex image processing algorithms – such as object segmentation and geometry estimation algorithms – that are very sensible to noise and may require controlled environments to ensure acceptable results. Pure geometry-based models such as 3D meshes are used in applications like computer games, Internet and movies. The rendering quality can be very high when the scene is computer generated. However, content creation is typically expensive and requires human assistance.

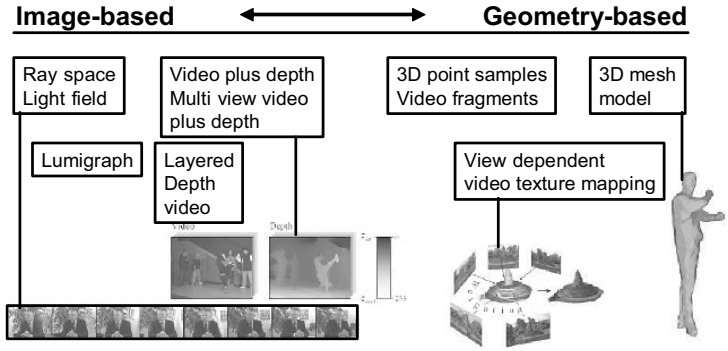


Figure 1.2: 3D scene representations [25]. Copyright Springer 2008.

Intermediate representations include both standard camera views and a geometric model. The most relevant one for 3DTV is the so-called Multi-view Video-plus-Depth (MVD) representation, which includes a number of camera views – typically between 2 and 8, but larger setups are possible too – and depth maps. As mentioned in Sec. 1.1, depth maps provide a description of the geometry of a scene from a specific viewpoint in the form of gray-scale images. Together with the corresponding texture images, depth maps allow creating 3D-like representations – often referred to as "2.5D" – that enable high quality view rendering, provided that the accuracy of depth information is good enough. The advantage of the MVD representation is that only a reduced number of views and corresponding depth maps is required. At the same time, depth images can be compressed very efficiently when specialized algorithms are applied thus avoiding significant bit rate increases when depth data need to be encoded and transmitted.

Different 3D representations imply different synthesis methods. Each single image-based view synthesis method can be seen as a particular way of resampling the light rays of the so-called *plenoptic function* [26]. The plenoptic function f of a 3D scene [27] is a 7-dimensional function that describes the intensity of all the irradiance observed at every point, coming from every direction, at any time instant:

$$I = f(x, y, z, \phi, \theta, \lambda, t), \tag{1.1}$$

where I is the light intensity at any 3D point (x, y, z) from any direction described by the spherical coordinates (ϕ, θ) for any wavelength λ and at

any time instant t. The process of synthesizing a virtual view corresponds to generating the light rays that define the pixels of the virtual image. Therefore, if the plenoptic function is completely known, the ultimate goal of 3DV communication can be achieved: the remote reproduction – in time or space – of the visual scene appearance from any viewpoint and at any time [22]. Unfortunately, the recording of a full plenoptic function is in general non feasible and would anyway result in a huge amount of data whose processing and transmission is implausible. For this reason, different capturing setups and image-based synthesis methods have been proposed to sample and approximate the plenoptic function in convenient manners.

Image-based rendering methods can be categorized according to three parameters: the amount of geometric information used, the spatial distribution of the acquired image data (the plenoptic samples), and the flexibility of the virtual viewpoints selection [26]. Figure 1.3 shows a classification of different rendering methods according to this categorization. Methods in which geometry is not explicitly represented are on the left side while the ones in which 3D models are exploited are on the right side. Methods requiring sparse camera setups are placed in the bottom part opposed to those requiring dense camera setups that are placed in the top part. The "freedom" in terms of virtual viewpoint selection (or virtual camera motion) is described by colors: light gray represents unrestricted selection, black indicates discrete selection, and dark gray intermediate solutions. As it can be noticed, depth map based representations combine the advantages of requiring not too dense camera setups (e.g. reduced memory demand and not too complex data acquisition) and using only partial geometric models (which allows compensation of the scene parallax and handling of occluded regions without requiring complex extraction) while providing full flexibility in terms of virtual viewpoint selection. For more efficient handling of occlusions the concept of Layered Depth Image (LDI) was introduced [28]. LDIs are multi-valued depth images in which multiple depth values are associated to pixels to describe different depth layers in the scene. The advantage of such representation is that both foreground and background objects are described entirely thus allowing reducing occlusion-related synthesis issues. Even though some acquisition setups able to acquire LDI data have been demonstrated [29], the LDI format is not as popular as the tra-

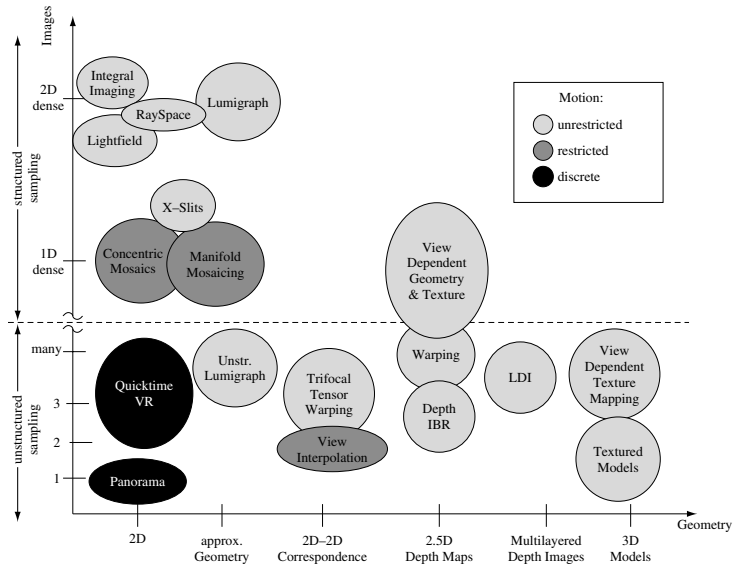


Figure 1.3: Classification of image-based rendering methods [26]. Copyright Wiley 2005.

ditional depth-based MVD format due to the need of more sophisticated processing algorithms.

1.2.2 Depth-based 3D Warping

As mentioned in the previous Sections, depth maps are used to perform the so-called "depth-based 3D warping" procedure, which allows projecting an input view to a different viewpoint. 3D warping is the base of any depth-based view synthesis algorithm, whose performances are crucial in DIBR systems. The warping equations are derived from the well-known "Pinhole camera model", which has been thoroughly studied in the literature [30, 31]. An extended camera model that takes into account more realistic devices considering effects like radial and tangent distortions for more accurate projections has also been proposed [32].

In order to enable the 3D warping some pieces of information are required [30]: the two extremes defining the depth range represented by the depth map (usually referred to as *near plane* and *far plane* values), the function that quantizes and maps physical distances into discrete values (or its inverse), the camera *intrinsic parameters* (including focal

length, pixel sizes, and image centre coordinates), and the camera *extrinsic parameters*, i.e. position and orientation of the camera with respect to a reference coordinate system. Intrinsic and extrinsic parameters of the target (virtual) camera are also needed in order to set the synthesis procedure. All these pieces of information – except for the virtual camera parameters – need to be sent to the receiver when view synthesis is performed at the decoder/display side. In order to facilitate the encoding of these data, the MPEG defined the MPEG-C Part 3 standard ("Representation of Auxiliary Video and Supplemental Information") [33] in 2007 for the single-view video-plus-depth format. The transport is specified in a separate specification called "Carriage of Auxiliary Data" [34].

The warping procedure operates at a pixel level by projecting each single pixel of the to-be-warped image into a 3D-point, and reprojecting the 3D-point to the target (virtual) image (*forward warping*). Due to the discrete nature of pixel coordinates – which implies the usage of rounding operations on the warped pixel coordinates – and object occlusions, not all the pixels in the warped image are filled. Moreover, multiple pixels in the input image can be projected on the same pixel location in the target image. To ensure a correct rendering of the foreground, the candidate pixel with the smallest distance from the camera plane is chosen ("Z-buffer" algorithm). If the depth data of the target view are available, *backward warping* can be performed: pixels in the target image are projected back to the input image and filled with warped pixel values. In this way no rounding operations are required in the target image, but the issue of the occlusions remains. However, occlusions can be filled properly if two or more input images, with corresponding depth maps, are available: occlusions from one view can be filled with data from other views.

For the case of stereoscopic video-plus-depth, the MPEG developed the so-called View Synthesis Reference Software (VSRS) [35, 36]. VSRS allows synthesizing any view in between the two input views and features advanced processing algorithms to reduce synthesis errors due to noise or errors in the depth data. VSRS uses both forward and backward warping: first the depth map of the virtual view is estimated from the input depth maps using forward warping, then the obtained depth map is used to synthesize a virtual view from each input view (backward warping), and finally the two virtual views are merged and conveniently

filled.

A number of algorithms have been proposed in the literature to improve the view synthesis performance in DIBR systems. Rana and Flierl [37] proposed a method for depth consistency testing to reduce the impact of depth errors in synthesized views. Sun *et al.* [38] analyzed the reliability of depth data to develop efficient blending and hole-filling algorithms. Another approach based on warping reliability has been proposed by Yang *et al.* [39] to reduce the video bit rate while enhancing synthesis performance. Post processing of virtual views based on adaptive Wiener filtering has been proposed by Yuan *et al.* [40] as a way to reduce the effects of compression artifacts on synthesized views. Filtering techniques have also been applied to depth data to solve the problem of recovering disoccluded regions in the synthesized view [41]. Advanced hole-filling methods exploiting depth information [42–44], adopting a hierarchical approach [45], or using texture synthesis techniques [46] have been proposed too, showing promising quality improvements. Different efficient and robust view synthesis methods have also been presented recently [47–50], confirming that the interest in DIBR systems is very high. Among all the methods, fast synthesis solutions have also been explored [51].

1.2.3 Depth Acquisition and Estimation

The acquisition of depth data plays an important role in a 3DV communication system based on the DIBR paradigm. The accuracy of depth data directly impacts the performance of any view synthesis algorithm, with consequences on the overall user experience. Acquisition of video data is usually less problematic as current video cameras can be used for most of the 3DV representations and rendering methods (see Fig. 1.3), as long as accurate positioning and synchronization can be done. For the case of depth data, special sensors are required if geometric information has to be directly acquired from the 3D scene. The most common depth acquisition techniques are active techniques, i.e. techniques in which light is both emitted and captured. Active techniques have the advantage that they can be used in almost any environment, however they require that objects in the scene have a minumum reflectance and present Lambertian surfaces, i.e. surfaces that diffuse incident light in the same wavelength

region of the illuminating light [52].

The most popular active techniques can be grouped in two categories: triangulation-based and Time-of-Flight (ToF). In a triangulation-based method a laser plane is projected onto the object to be acquired marking a profile line of its local shape. According to the position and deformation of the profile line, a camera can infer the geometric properties of the illuminated portion of the object with great precision [52]. Multiple acquisitions – using more than one camera and/or a scanning mechanism – are needed in order to acquire an entire object, which make this approach suitable for static scenes. ToF techniques are based on a different principle: distances are calculated by measuring the time-of-flight between an emitted signal and its corresponding echo [53]. Two main categories are identified among the ToF sensors: those based on amplitude or frequency wave modulation and those based on pulsed wave or time delay. The choice of a ToF sensor depends on the specific application, e.g. precision requirements, acquisition range, and budget [52]. Focal plane arrays are a promising evolution of the ToF sensors: they use wider sensors able to perform parallel measurements of the light field emitted by the system emitters and reflected by the objects in the scene. Focal plane arrays operate at video rates (up to 50 frames per second) and are thus suitable for dynamic scenes. However, they usually do not support high spatial resolutions and their performance varies with the specific illumination conditions [54, 55].

Among all the different depth cameras, the Microsoft Kinect$^{\text{TM}}$ range camera is worth a special mention due to its recent success and wide popularity. The Kinect$^{\text{TM}}$ camera includes an infrared projector used to project infrared light coded patterns on the scene, and an infrared camera able to capture the projected patterns. The geometry of the scene can be estimated from the deformations of the projected patterns [56]. The Kinect$^{\text{TM}}$ camera operates at 30 frames per second providing depth images with VGA spatial resolution (640×480) and covering a nominal depth range of about 15 meters. Due to the fact that the two infrared devices are placed one next to the other, the occlusions issue appears. Areas for which a depth value can not be estimated are assigned a zero value. Zero values may appear also because of noise in the infrared pattern projection or capturing steps, and represent a challenge for the various depth processing algorithms. An example of a depth map acquired

Figure 1.4: Example depth image from Microsoft Kinect™ [60]. Copyright IEEE 2011.

by a Kinect™ camera is provided in Fig. 1.4. The Kinect™ camera is nowadays used not only for gesture recognition applications [57], but also for 3D scene reconstruction [58], augmented reality, robotic navigation, and other applications [59].

Depth information can be also estimated from texture data. By matching corresponding points between two or more views of the same 3D scene, depth and disparity data can be inferred. The problem is more challenging than it may look and many different methods for robust stereo and multi-view matching have been proposed in the literature in the last decades. A review of stereo algorithms is out of the scope of this thesis; the interested reader is referred to [61, 62]. The method proposed by Zitnick *et al.* [17] is of particular relevance for this thesis as the MVD sequences presented in their work – namely *Breakdancers* and *Ballet* – have become reference test sequences in the fields of 3D and multi-view video coding based on DIBR. An example of a view and the corresponding depth map from the *Ballet* dataset is provided in Fig. 1.1. Depth data are obtained using a texture segment matching algorithm and specialized filtering and refinement techniques.

Together with the VSRS, the MPEG 3DV also developed and implemented the Depth Estimation Reference Software (DERS) [63], a useful reference tool for depth estimation from multi-view video data. DERS employs block-matching and post-processing algorithms to estimate depth data. A number of optimizations and alternative solutions have been proposed to improve the performance of DERS, like for instance [64], in which optical flows are considered.

1.2.4 Compression and Transmission of 3DV

Efficient encoding and transmission of 3D and multi-view video has been an active field for both researchers and standardization organizations for many years. As mentioned in Sec. 1.1, the encoding of multi-view video was supported already in the H.262/MPEG-2 Video standard [4] with the MPEG-2 Multiview Profile [3] introduced in 1996. The most recent international standard for the compression of multi-view data is the MVC extension of the H.264/MPEG-4 AVC standard (also referred to as MPEG-4 Part 10 Amendment 4). H.264 MVC [6, 7] includes a so-called "base view", which is coded independently from the other views so that single-view H.264/AVC decoders can extract a 2D version of the content from the bit stream. Inter-view prediction is enabled to exploit redundancy between different views in addition to temporal redundancy. Adaptive prediction is used: the choice of the best predictor between temporal and inter-view references is made on a block basis by minimizing a Rate-Distortion (RD) cost function. Even though inter-view prediction is allowed only between views at the same time instant, MVC features a very flexible management of the reference pictures, following the design of H.264/AVC and introducing only minor changes to the high-level syntax. Such flexibility allows multiple temporal/inter-view prediction configurations to fit the specific coding requirements, e.g. performance, delay, random access, and memory. MVC showed an average bit rate reduction of 20% compared with the simulcast total rate for the case of videos with up to 8 views. In stereo setups average rate reductions of 20%-30% for the non-base view were reported, but in both stereo and multi-view cases performance strongly depend on the specific prediction structure adopted. More details on MVC, including profiles and levels, can be found in [7, 65, 66].

When the interest in the MVD format became significant, the problem of efficient mono-view and multi-view depth video coding also gained interest. 8-bit depth maps – the most common format – can be interpreted as luminance-only video signals and can be thus encoded by any state-of-the-art video codec. Smolic *et al.* [25] report that depth information can be compressed very efficiently when standard video codecs are used, and roughly 10%-20% of the bit rate necessary to encode texture video is usually enough to encode depth data at a good quality. However, it has been shown that H.264 MVC is not optimal for the compression

of multi-view depth video in terms of view synthesis performance [67]. Recent studies show that the optimal texture-depth rate partitioning is content-dependent [68] and influenced by some key characteristics of the sequence, including baseline distance and contrast of neighboring background/foreground regions [69]. Some solutions for optimal joint texture-depth bit rate allocation have been proposed, e.g. [70] in which a synthesized view distortion estimation model is exploited to properly allocate the rate, and [71] in which combined texture-depth motion encoding and an activity-based rate allocation strategy were proposed.

When the MPEG issued the "Call for Proposals on 3D Video Coding Technology" [72] in 2011, more than 20 contributions were received. The proposed technologies have been grouped into two main categories: those compatible with the current H.264/AVC standard and those compatible with the upcoming H.265/High Efficiency Video Coding (HEVC) standard [73, 74], recently promoted to Final Draft International Standard (FDIS) status [10]. Some hybrid architectures have been proposed too. Most of the contributions were based on the MVD data format and many new coding tools were proposed to improve the performance of both texture and depth coding, including a number of solutions for edge-aware depth coding [75]. Examples of depth coding tools include a texture-driven skip mode for multi-view depth video [76], a new intra prediction mode in which planar approximations are defined [77], and down-/up-sampling strategies followed by specialized filtering algorithms [78]. Standardization activities are currently well underway and core experiment workplans are being finalized [79].

In parallel with the standardization process, the research community is proposing an impressive amount of new solutions and schemes for efficient coding of 3D and multi-view video, with particular focus on depth compression. Kim *et al.* [80, 81] and Zhang *et al.* [82] proposed to analyze the effects of depth compression on synthesized views to improve depth coding reporting promising performance improvements. Platelet-based depth coding [83, 84] was proposed as a way to preserve steep discontinuities in the depth signal: depth maps are modeled using piecewise-linear functions (platelets) within blocks of variable sizes defined using a quadtree decomposition. The method was later combined with H.264/AVC-like intra prediction and a dictionary-based approach by Lucas *et al.* [85] to improve the compression performance.

Graph-based edge-adaptive transforms for depth coding have been proposed by Shen *et al.* [86] for 4×4 blocks as an alternative to the standard Discrete Cosine Transform (DCT) used in H.264/AVC, which fails to represent sharp discontinuities efficiently (i.e. with few non-zero coefficients). The idea is to define a graph structure among pixels in edge blocks so that pixels on different sides of the edge are not connected, and use this structure as base for the transform. The method was subsequently improved with more efficient definition and coding of the graph structures [87]. Cheung *et al.* [88] proposed a completely different approach introducing the concept of *don't care region*: depth values can be conveniently manipulated within a suitable range to obtain a sparse depth representation and improve the coding performance of a JPEG encoder without affecting the quality of synthesized views. Another approach based on similar concepts was developed by Zhao *et al.* [89] with good results in terms of depth coding. The formulation in [88] was extended in [90] where more general scenarios were addressed, and combined with the concept of graph-based transforms previously introduced to achieve further improvements [91].

Other approaches for depth map coding include – not exhaustively – pattern matching techniques [92], edge block binary approximations integrated within the H.264/AVC framework [93], efficient depth segmentation and down-sampling algorithms for edge preservation [94], depth down-/up-sampling exploiting texture data [95], and specialized filtering techniques [96]. Wavelet-based depth coding has also been explored [97], including techniques based on wavelet transforms on graphs [98]. Scalable solutions for depth images have been recently proposed to improve the performance of a JPEG2000 encoder around sharp discontinuities while maintaining its scalable properties [99, 100]. Methods for error concealment of MVD data [101], efficient transport strategies [102] and Multiple Description Coding (MDC) approaches for 3DTV [103] have been analyzed too.

Lossless compression of depth data has also been addressed in the literature. Lossless coding ensures that no data loss occurs in the encoding/decoding process allowing for very precise processing operations at the decoder side. This can be of interest not only in DIBR scenarios, but also when depth images are used for purposes such as scene matting, object and face tracking. Recent results show that differently from the

case of natural images in which the average lossless compression factor is between 2 and 4, depth map compression can provide lossless compression factors of 50 and above – for both still images and videos – when specialized algorithms are considered. Kim *et al.* [104] proposed a bit-plane-based coding method exploiting inter-bit-plane redundancies and motion estimation. A scalable lossy-to-lossless solution based on the cellular automata model was proposed by Cappellari *et al.* [105]. Solutions based on minimum description length segmentation [106] and a modified structure of the H.264/AVC Context-based Adaptive Binary Arithmetic Coding (CABAC) [107] were also proposed. Finally, a near-lossless and low-complexity scheme for Kinect-like depth maps was presented in [60].

1.2.5 Research Trends

Following [108], the current research trends can be summarized by briefly reviewing scopes and activities of some of the research consortia in the 3D, Immersive, Interactive Media (3DIIM) Cluster [109], the main organization embracing EC funded projects on 3D and holographic media capture, distribution, representation, and other aspects of immersive and interactive media communication.

3DTV (2004 – 2008): The technical focus of the project was on 3DV communications, including all the system components such as capture, representation, coding, transmission, and display, with a particular focus on Internet-friendly technology development. Among the technical goals that have been achieved are: the confirmation that multiple synchronized video cameras are the most promising capturing technology for 3DV, the development of an end-to-end platform for the streaming of 3DV over IP, the analysis of human factors related to autostereoscopic displays, significant contributions in the development of the MPEG-C Part 3 specification (see Sec. 1.2.2), H.264 MVC, and the upcoming MPEG 3DV standards [110].

MUTED (2006−2009) and **HELIUM3D** (2008−2010): these projects focused on advancing the state-of-the-art of 3D display devices supporting multiple mobile viewers simultaneously and natural user interactions with the 3D content [111, 112].

MOBILE3DTV (2008 – 2010): Robust delivery of stereoscopic video over Digital Video Broadcasting – Handheld (DVB-H) channels was the main focus of the project, including the development of error resilience and error-concealment capabilities for handheld devices. Different video content formats – including video-plus-depth – and video codecs suitable for 3DTV delivery have been analyzed, and metrics for subjective quality assessment of stereoscopic video have been developed [113]. Similar targets were addressed in the **3DPHONE** project (2008 – 2011), with a stronger focus on the identification of hardware and software specifications and requirements for the handling of 3D data [114].

3D4YOU (2008 – 2010): This project proposed a new 3D delivery format based on LDIs with specialized occlusion handling procedures. A novel acquisition setup based on the proposed format was also demonstrated [115].

3D Presence (2008 – 2010): The goal of this project was to implement a 3D videoconferencing system able to provide to the participants the feeling of physical presence, including eye contact and gesture-based interaction. A 3D multi-perspective multi-screen videoconferencing system over the Internet with virtual 3D scene rendering capabilities has been demonstrated [116].

2020 3D Media (2008 – 2012): The project focused on capture, production, delivery and display of 3D audio and video content for immersive entertainment purposes in both private (home) and public scenarios. Technical achievements include the development of new 3D camera architectures targeting the video-plus-depth format, standard contributions in the field of MVC, development of algorithms for automatic 2D-to-3D content conversion, and design of error concealment methods for multi-view data [117].

DIOMEDES (2010 – 2012): This project focused on the development of a hybrid 3D P2P-DVB distribution system for the delivery of multi-view video and multi-channel audio for entertainment purposes. The system is based on a scalable multi-view video coding architecture and exploits multi-view depth data for real-time

free-viewpoint rendering of 3D content. Particular focus was devoted to the development of solutions that take into account perceptual quality and the design of advanced depth estimation techniques [118].

MUSCADE (2010−2012): The MUSCADE project targeted the entire 3DTV chain, from acquisition to display. The proposed architecture featured a 4-view High Definition (HD) acquisition system including 3D audio capture, scalable video coding solutions based on the MVD format, advanced error resilience and concealment techniques, and congestion control management for IP environments suitable for a multitude of network platforms and transmission channels [119].

SkyMedia (2010 − 2012): This project aimed at developing a new end-to-end multimedia architecture able to provide immersive media experiences during live events in open environments. Sophisticated unmanned aerial capturing devices have been designed and implemented. Scalable coding solutions based on both H.264 MVC and depth-enhanced formats have been considered [120].

FascinatE (2010 − 2013): FascinatE's goal is to allow users to interactively view and navigate around an ultra-high resolution panoramic video showing a live event, with particular focus on real-time view rendering and display-dependent content adaptation. Multiple HD cameras and microphones are used as acquisition devices. The proposed system does not capture nor build a 3D representation of the scene and only relies on multi-view data to render very high-quality views [121].

ROMEO (2011 − 2014): This project aims at delivering live multi-view 3DV to a variety of users exploiting both broadcast and IP networks under the real-time constraint. The MVD format is exploited to allow flexible view rendering capabilities at the end-user side and enable e.g. FVV browsing and depth perception adjustment [122].

SCENE (2011 − 2014): The SCENE project targets the delivery of rich 3D media experiences by combining multi-view video-based scene representations with computer graphics modeling. MVD data are

captured, processed and retargeted exploiting hybrid scene representations, and optimized for network delivery and flexible rendering on a variety of display platforms [123].

REVERIE (2011 – 2015): The REVERIE project aims at creating immersive collaborative environments supporting inter-personal communication of remotely located users exploiting avatar-based interactions and highly-realistic 3D modeling based on the acquisition of MVD data. Scalable content-aware networking and real-time distributed scene rendering are among the main focus points [124].

As a general comment one can notice that a number of recent and ongoing research projects devote particular attention to the video-plus-depth format or other depth-enhanced video formats. Significant improvements have been achieved for the occlusion handling issue and the usefulness of depth information for future 3DV communication systems seems to be gaining a wide acceptance.

1.2.6 Depth-enhanced Distributed Video Coding

Distributed Video Coding (DVC) [125, 126] is a novel coding paradigm that aims at providing a flexible distribution of the computational complexity between encoder and decoder without penalizing compression efficiency. In standard hybrid coding architectures most of the complexity lies at the encoder side where spatial and temporal redundancies are exploited to maximize the compression performance. Such asymmetric scheme is suitable for broadcast and down-link applications, in which the video content is encoded once and decoded many times. However, in a number of scenarios – like video surveillance and wireless camera networks – low-complexity encoders are highly desirable due to strict constraints in terms of device size and/or power consumption. More relaxed constraints are usually set at the decoder, which can possibly include computationally heavy routines. In these scenarios DVC appears as a promising solution as it allows performing motion estimation – the most computationally intensive procedure – at the decoder side. The DVC paradigm is based on two theoretical results: the Slepian-Wolf theorem [127] and the Wyner-Ziv theorem [128]. The former shows that

the joint decoding of two statistically dependent signals independently encoded in a lossless manner requires the same rate as for joint encoding and decoding; the latter extends the result to the lossy coding case.

Distributed coding solutions have been analyzed also for 3DV based on the multi-view [129], video-plus-depth [130], or MVD [131] formats. Distributed coding of multi-view or mono-view video-plus-depth data is of interest not only for 3DV, but also in scenarios in which depth data allows for more precise and robust processing, such as video surveillance, object tracking, activity detection, and scene matting. Due to the low-complexity/resource-constrained encoding devices used in DVC settings, inter-camera communication is typically not feasible. Therefore, for the case of distributed coding of video-plus-depth data not only the temporal correlation but also the inter-component correlation has to be exploited at the decoder side, where advanced joint texture-depth decoding strategies might provide higher RD performance.

1.3 Goals of the Thesis

The main goal of this thesis is to advance the state-of-the-art in the field of 3DV coding, with particular focus on depth coding for DIBR-based 3DV communication scenarios. If on one hand the huge expertise developed in the field of monoscopic video coding remains useful and relevant when dealing with multi-view and 3DV coding, on the other hand current 3DV formats introduce new components, such as depth maps, which require novel specialized coding strategies for optimal compression. This thesis explores new solutions for the problem of efficient coding of depth maps exploiting their peculiar features, such as wide smooth regions and sharp edges, and their correlation with the corresponding texture component. The target is not only to improve the compression performance of depth data, but to improve the overall coding performance of DIBR-based 3DV codecs taking into account the view synthesis process, in which the accuracy of depth data plays a central role.

Another goal is to design and develop new approaches for efficient distributed coding of video-plus-depth data, with particular focus on the generation of accurate frame estimates at the decoder side that exploit the correlation between the texture and depth signals. The scenario of interest addresses the distributed coding of a video-plus-depth stream

captured from a standard video camera and a co-located depth camera.

1.4 Structure of the Thesis

This thesis is structured as follows: Chap. 2 describes the main contributions highlighting their relevance in the context of efficient multi-view and 3DV coding. Chap. 3 briefly summarizes the achievements of the published material included in the thesis, and finally Chap. 4 summarizes the main results and outlines future development directions in the field of 3DV communication.

Chapter 2

Novel Coding Schemes for Multi-View 3D Video using Depth Maps

The contributions of this thesis can be divided into two groups. The first one includes new approaches for lossy and lossless depth map coding for Three-Dimensional Video (3DV) systems based on Depth-Image-Based-Rendering (DIBR). Solutions based on both standard and non-standard approaches are explored, including segmentation-based and edge-aware algorithms on the lossy side, and a depth-warping-based scheme on the lossless one. Experimental results show that significant performance improvements can be obtained when the specific features of depth images are taken into account. In the second group new schemes for distributed video-plus-depth coding are described, showing how the motion correlation between texture and depth data can be exploited to improve compression efficiency. Advanced motion estimation techniques are used to capture motion information from one component to compensate the second one thus generating different estimates of the frames being decoded. When conveniently fused, inter- and intra-component estimates provide more accurate estimates for a variety of coding setups, reflecting in improved overall Rate-Distortion (RD) performance.

2.1 Compression of Depth Information

2.1.1 Lossy Depth Map Coding

Joint Multi-View Plus Depth Image Coding based on 3D Warping

Multi-view image coding has been a research field of wide interest in the last decades. The problem of efficient compression of multi-view data has been approached in many different ways, but ultimate solutions have not been found as the broad variety of acquisition setups and conditions makes it very difficult to identify a universal approach. Particular interest is devoted to 1-D and 2-D camera arrays, the most common setups for 3DV. For the case of 1-D camera arrays, depth/disparity-enhanced 3D representations have also been considered. Depth maps can be used as a mean to capture the redundancy among multiple views and allow improving compression performance. Multi-view image coding based on disparity compensation has been explored in the literature and a framework showing the benefits that depth data can provide in terms of multi-view image coding is presented in [132] for the case of 8-view camera settings. In this scheme depth maps are used to warp the different views to a reference viewpoint, i.e. the one of the central view. Each warped view can be seen as a prediction of the central view made from a different viewpoint. Warped views are stacked in a 3D data structure and – after a hole-filling step – encoded by means of block-based 3D Discrete Cosine Transform (3D-DCT), quantization, and entropy coding. Additional information for resolving occlusions is also included in the bit stream. At the decoder side, views are jointly decoded, warped back to the original viewpoints, and occlusions are filled. The algorithm shows that significant improvements – especially at low rates – compared with a standard H.264 MVC codec can be achieved. However, the uncompressed multi-view depth map is supposed to be available at both encoder and decoded sides, which makes the scheme not suitable for practical implementations. In **PAPER 8** a more feasible solution is proposed. In this scheme only the central depth map is required for the warping operations at the encoder and decoder, and depth map coding is included too. Compared with [132], the 3D-DCT based coding of the stacked views has been replaced by a standard H.264/AVC coder (see

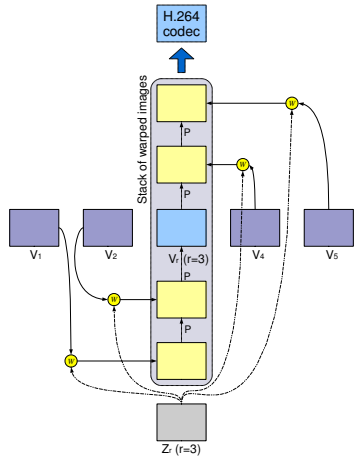


Figure 2.1: Stack creation from a single depth map and relative encoding from **PAPER 8**. In this case view number 3 is the central view of a 5-view image and its corresponding depth map is used to warp the non-central views (see labels W). Copyright ACM 2010.

Fig. 2.1 for a representation of stack creation and encoding). In this case warped images are considered as consecutive frames in a video sequence. As the precision of warped views strongly depends on the accuracy of depth data, some misalignments may appear among warped views in the stack. Motion estimation and compensation are therefore used to capture misalignments and allow for an efficient coding. In the proposed scheme depth data are compressed with a standard H.264/AVC Intra coder. The Quantization Parameters (QPs) for depth and texture are chosen so that about one fourth of the total bit rate is allocated for depth data, inspired by [25]. Experimental results show that the proposed method can outperform H.264 MVC by up to 1.2 dB in terms of Peak Signal-to-Noise Ratio (PSNR) at low rates on the considered test sequences. At high rates the proposed method is unable to provide satisfactory performance because the 3D warping algorithm introduces inaccuracies in the warped images even when uncompressed depth data are used. **PAPER 8** describes a realistic scheme for multi-view image coding. The scheme shows how a single depth map can be beneficial for coding purposes in a multi-view scenario, including settings with 2 and 8 cameras.

Edge-aware Depth Coding

As broadly discussed in Chap. 1, edge-aware depth map coding is a research field of high interest. The goal is to develop efficient coding methods able to preserve sharp discontinuities in the depth signal, of crucial importance for DIBR algorithms. Particularly, edges in the depth signal that correspond to edges in the texture separating between two or more objects are the most important ones. In **PAPER 7** a new algorithm for edge-aware depth coding exploiting segmented color data for the case of single-view plus depth still images is presented. The novel idea is to exploit texture segments to predict shapes of corresponding surfaces in the depth signal. If texture segmentation is done properly, segments correspond to different objects in the scene, or relative portions, and thus identify the most important discontinuities to be preserved. Basing on the assumption that depth images consist of smooth regions corresponding to surfaces in the scene divided by sharp edges, a depth segment – i.e. texture segment applied to depth data – can be approximated with a parametrized planar surface. The advantage is that for each depth segment only the corresponding geometric parameters need to be transmitted to the decoder, allowing for significant bit rate savings: segmentation information can be in fact generated at the decoder side from reconstructed texture. Improvements up to 2 dB in terms of depth PSNR are reported. Figure 2.2 shows an example of texture segmentation. After the first segmentation step, segments are partitioned into smaller regions to improve the performance of the depth surface fitting algorithm.

A different approach for depth coding that does not exploit corresponding texture information is presented in **PAPER 3** and **PAPER 1** for the case of depth video. The algorithm is based on a standard H.264/AVC codec and introduces a new Intra mode specifically targeted to depth macroblocks containing arbitrarily-shaped edges. These blocks are typically not represented well by the standard modes of H.264/AVC Intra as the Discrete Cosine Transform (DCT) is not efficient in case of signals presenting sharp discontinuities (i.e. many non-zero high frequency transform coefficients are usually obtained). Therefore, for these blocks high RD costs are obtained and sharp edges are not preserved well. Based on the observation that at a local scale edge regions typically separate between two depth values only, a new mode is proposed.

(a) (b) (c)

Figure 2.2: Example of texture segmentation for the *Breakdancers* sequence from **PAPER 7**: (a) reconstructed texture, (b) segmented texture, (c) segmented and partitioned texture. Copyright IEEE 2011.

The algorithm operates at a full macroblock level and groups pixels into two partitions – described through a binary mask – according to the edge structure. Each partition is then approximated by a flat surface. As discussed in Chap. 1, other methods approximating depth maps at a local level with flat or planar surfaces have already been proposed in the literature. However, in these methods the encoding of edge information is typically done independently block by block, basing only on local edge features. In the proposed algorithm, portions of long edges spanning multiple adjacent macroblocks can be encoded jointly to allow significant bit rate savings while preserving the original edge sharpness. This result is achieved by using context-based arithmetic coding to encode edge information (i.e. block partitioning) at a macroblock level. Context statistics from previously encoded edge macroblocks are exploited to improve the compression performance. The concept is illustrated in Fig. 2.3. Experimental results in **PAPER 3** show that the proposed mode can provide promising performance improvements compared with standard H.264/AVC Intra coding in terms of depth quality and view synthesis performance using reconstructed depth data: average Bjøntegaard bit rate reductions of about 12% and 25% are noticed, respectively. In **PAPER 1** a much broader comparison against other state-of-the-art edge-aware depth coding methods is provided, highlighting the benefits of the proposed method. Differently from **PAPER 3**, in which the proposed mode is enabled in I slices only, in **PAPER 1** the proposed mode is enabled also in P slices, allowing efficient encoding of edge information also in predicted frames.

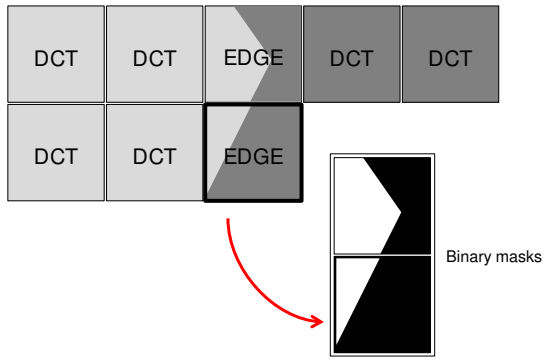


Figure 2.3: Proposed inter-block binary mask coding: if a previously encoded neighboring macroblock was encoded with the proposed mode, its binary mask can be attached to the one of the current macroblock being encoded, and corresponding context statistics exploited to initialize the context-based binary encoder.

Other approaches for lossy depth coding have been explored during the Ph.D. studies in collaboration with other groups, namely graph-based edge-aware lifting transforms for depth blocks featuring arbitrarily-shaped edges as a way to reduce the complexity of the graph-based transforms presented in [87]; edge-based soft decoding strategies for depth data down-sampled at the encoder side and up-sampled at the decoder side, inspired by the edge-based perceptual image coder presented in [133]; and inpainting techniques based on Partial Differential Equations (PDEs), motivated by promising results reported for cartoon-like images [134]. However, none of the listed approaches turned out to be competitive with standard codecs for the considered setups.

Performance Comparison

The proposed coding schemes were evaluated considering different comparison setups. In **PAPER 8** the quality of decoded multi-view images was evaluated while in **PAPER 7** the PSNR of decoded depth images was studied. **PAPER 3** considered both decoded depth PSNR and synthesized view PSNR, and finally **PAPER 1** evaluated decoded depth PSNR, synthesized view PSNR versus depth bit rate and texture plus depth bit rate, and convex hulls defined by synthesized view PSNR and texture plus depth bit rate. A direct comparison among all the algorithms is not possible, but **PAPER 7** and **PAPER 3** can be compared

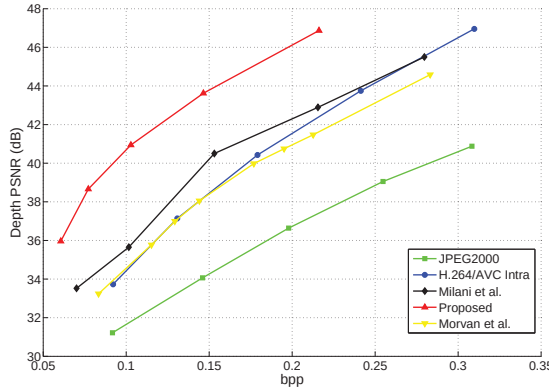


Figure 2.4: RD performance comparison on the *Teddy* depth image from **PAPER 3**. Copyright IEEE 2013.

in terms of depth PSNR on the *Teddy* depth image (shown in Fig. B.23), as done in **PAPER 3** and reported in Fig. 2.4. The comparison is interesting because it involves one algorithm in which texture is used to improve depth coding and one in which no inter-component correlation is exploited. As it can be noticed, the edge-based scheme (red line, **PAPER 3**) outperforms the segmentation-based method (black line, **PAPER 7**) at all the considered bit rates, marking an average PSNR improvement of about 4 dB. Improvements are even bigger when compared with the other methods reported. The reason for such a difference in performance can be found in the fact that the considered depth image features sharp and clean edges, which are encoded very efficiently by the proposed edge-aware intra mode. Moreover, the great amount of details that the texture exhibits – especially in the background area, but also in foreground regions – may cause the segmentation algorithm to produce an over-segmented texture, which makes it hard to predict surface shapes in the depth signal with sufficient accuracy. This comparison shows that an appropriate modeling of the signal being encoded can help more than the availability of correlated data.

2.1.2 Lossless Depth Map Coding

Lossless coding of depth maps is of interest not only because it allows avoiding view synthesis artifacts due to depth compression, but also because it enables accurate processing operations at the decoder side for

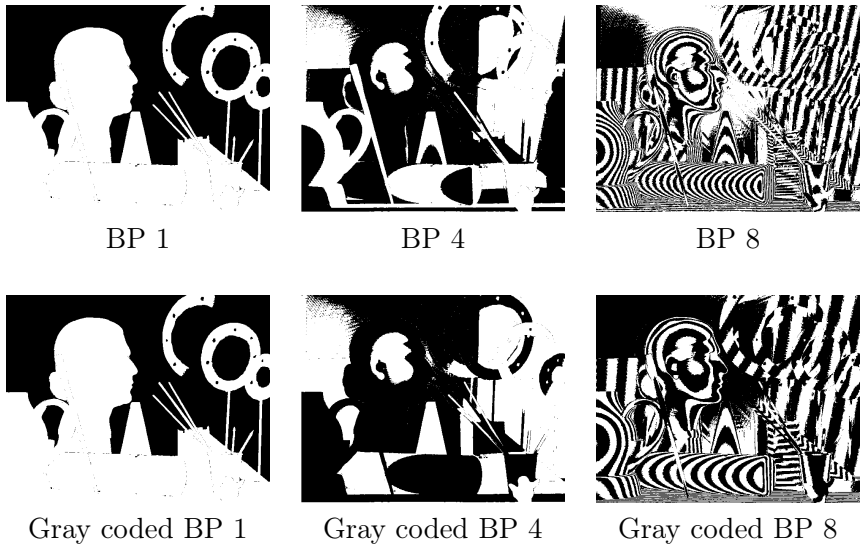


Figure 2.5: Gray coding of depth bit-planes (BPs). Top, left to right: original depth bit-plane 1 (most significant bit-plane), 4, and 8 (least significant bit-plane) from the Middlebury *Art* dataset [135]. Bottom, left to right: Gray coded bit-plane 1, 4, and 8. Note how Gray coding – which corresponds to the calculation of the *exclusive or* between adjacent bit-planes – allows increasing the local correlation of bits in bit-planes 4 and 8 compared with the corresponding original bit-planes.

a number of different purposes, as mentioned in Chap. 1. This thesis includes a contribution in the unexplored field of lossless compression of stereoscopic depth maps. If on one hand very efficient lossy coding of mono-view and multi-view depth maps can be achieved as discussed in previous Sections, on the other hand it has not been shown that lossless mono-/multi-view depth map coding can also be performed very efficiently when specialized algorithms are considered. In a coding scenario for stereoscopic depth/disparity maps the main challenge is to exploit inter-view correlation in an effective manner. In **PAPER 6** a novel solution for lossless coding of stereo disparity maps is proposed. The algorithm is based on bit-plane decomposition, Gray coding, and context-based arithmetic coding with adaptive template selection. Gray coding is used to increase the local correlation of bits in each bit-plane (see Fig. 2.5 for a visual example), which favors the context-based coding.

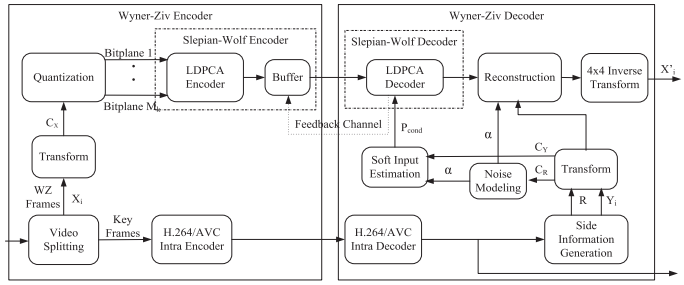


Figure 2.6: Block scheme of the DVC coded presented in [136]. Copyright Elsevier 2012.

While the left map is intra-coded, depth warping is used to predict the right map, and the prediction is used to provide a broader selection of candidate template pixels for the encoding of the right map. Together with very promising results in terms of compression efficiency (bit rate reductions between 22% and 76% compared with standard lossless codecs are reported), the proposed approach features a progressive representation of depth data and a structure suitable for parallel computing, both relevant characteristics for practical implementations.

2.2 Distributed Approaches for Video-plus-Depth Coding

In distributed video-plus-depth coding, inter-component correlation can be exploited at the decoder side to improve the RD performance, as discussed in Sec. 1.2.6. Since texture and depth represent two different aspects of the same physical 3D scene, the two components exhibit a high degree of correlation [71], especially in terms of motion behavior. This thesis includes a number of contributions in the field of distributed video-plus-depth coding for the cases in which one component is exploited to improve the Wyner-Ziv coding of the other one. All the proposed solutions are based on the state-of-the-art codec presented in [136] (see Fig. 2.6). In **PAPER 5** a simple scheme able to exploit texture motion information to improve distributed coding of depth maps is presented. The novelty is a new setup in which Motion Vectors (MVs) calculated on texture data are used to motion compensate corresponding depth images and obtain an estimate of the current depth frame being decoded, re-

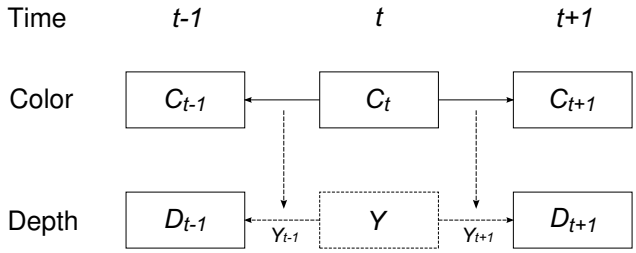


Figure 2.7: Depth SI generation algorithm from **PAPER 5**. Copyright SPIE 2013.

ferred to as Side Information (SI). In order to allow the calculation of the MVs, the texture frame at the same time instant of the depth map being decoded has to be available, together with the previous and following frames of both components. If this assumption is verified, Bjøntegaard PSNR improvements up to 1.88 dB compared with the state-of-the-art codec [136] can be obtained without affecting decoding delay and decoder complexity significantly. The scheme is illustrated in Fig. 2.7. In **PAPER 4** a more realistic scenario is addressed: texture data are also coded, and a reconstructed version is available at the decoder. Additionally, motion estimation techniques based on optical flow algorithms [137] are used to capture dense motion information from the texture. Then, MVs are used to motion compensate the depth signal, as done in **PAPER 5**. Together with the SI generated with optical flows, the proposed scheme includes two more SIs: one obtained using the so-called Adaptive Rood Pattern Search (ARPS) [138] algorithm to estimate texture motion and compensate the target depth frame, and a depth-only SI generated using the Overlapped Block Motion Compensation (OBMC) algorithm introduced in [136]. SIs are then combined in an efficient manner exploiting multi-hypothesis decoding techniques [139]. The availability of multiple SIs generated from different signals and with different techniques is beneficial in terms of accuracy of the decoded signal: if the fusion of the SIs is done properly, the most accurate portions of each SI are selected and included in the combined SI. Overall, the proposed inter-component scheme reports Bjøntegaard PSNR improvements between 1.50 dB and 4.95 dB compared with the intra-component decoder in [136].

In **PAPER 2** similar concepts are used for the "mirror" coding problem. In this case depth maps are exploited to improve the coding per-

formance of texture data. Multiple optical flow based SIs are generated, including depth-only, texture-only, and joint depth-texture SIs. Again, multi-hypothesis techniques provide efficient fusion of SIs, showing that the proposed decoder is highly robust against quantization noise in reconstructed depth maps. Moreover, the proposed method provides interesting Bjøntegaard PSNR improvements between 0.90 dB and 1.49 dB compared with the reference decoder [136], demonstrating that depth motion information can be effectively exploited to improve Wyner-Ziv decoding of texture data, if an increase of the computational complexity of the decoder is accepted.

Chapter 3

Description of Ph.D. Publications

In this Chapter an extended description of the publications included in the thesis is provided, highlighting novelties and main results. Papers are divided into two groups: compression of depth information and distributed approaches for video-plus-depth coding, following the subdivision in Chap. 2. Within each group, papers are presented in chronological order.

3.1 Compression of Depth Information

PAPER 8: A Joint Multi-View Plus Depth Image Coding Scheme Based on 3D-Warping

In this paper a novel scheme for joint multi-view plus depth image coding based on Depth-Image-Based-Rendering (DIBR) is presented. The work is inspired by the scheme presented in [132], in which multi-view depth maps are exploited to warp the different views to a common viewpoint. Warped views are then jointly encoded and warping is performed again at the decoder to render the reconstructed multi-view image. Differently from [132] in which uncompressed multi-view depth maps are supposed to be available at both encoder and decoder sides, the proposed scheme considers a more realistic scenario in which only one depth map is needed.

The encoding of depth information is also addressed. The proposed algorithm replaces inter-view prediction exploited in standard solutions for multi-view coding like H.264 MVC, with a view warping procedure: using *backward warping* the views are projected to the viewpoint of the central view to form a set of predictions of the central view from the various original viewpoints. Only the depth map of the central view is required for the warping operations. Warped views are stacked and occlusion regions are filled by means of linear interpolation from adjacent views. The stack is then encoded by means of H.264/AVC, considering warped views as consecutive frames of a video sequence. In case of accurate depth data no motion should appear among the warped views in the stack. On the other hand, noisy depth data cause edge misalignments and other artifacts in the warped views, which are captured by the motion estimation and compensation procedures of H.264/AVC. At the decoder, views are warped back to the original viewpoints using *forward warping*. Additional information to resolve occlusions is included in the bit stream. Depth data are compressed using a standard H.264/AVC Intra coder and corresponding Quantization Parameters (QPs) are selected in order to allocate about one fourth of the total bit rate to depth information. Experimental results on both stereoscopic and 8-view sequences show that the proposed method can outperform H.264 MVC at low rates. At high rates H.264 MVC achieves higher performance as 3D warping does not allow obtaining very high-quality synthesized views due to its sensitivity to depth inaccuracies. For the 8-view case the proposed method outperforms H.264 MVC at low rates even considering the additional rate for depth data, which is not required for H.264 MVC encoding/decoding. In Free Viewpoint Video (FVV) scenarios in which depth maps are anyway required for view synthesis at the decoder, the proposed method looks even more appealing as the gain over H.264 MVC increases when depth bit rate is included in both coding approaches.

PAPER 7: Efficient Depth Map Compression Exploiting Segmented Color Data

This paper introduces a novel scheme for depth coding exploiting

texture segmentation in a single-view plus depth coding scenario. A texture image is encoded using a standard H.264/AVC Intra coder. Reconstructed texture is segmented at the encoder side using a graph-based segmentation algorithm [140]. A second partitioning step is performed in order to reduce the size of segments exceeding a predefined size and ensure a higher fitting accuracy of depth surfaces in the next steps. Partitioned segments are then used to predict surface shapes in the depth image. Each depth segment – defined from texture segmentation – is approximated by a planar surface using a fitting algorithm based on the RANSAC paradigm [141]. If the fitting is accurate enough in terms of Mean Squared Error (MSE) for the target bit rate, surface coefficients are compressed by means of Huffman coding. If the planar approximation is not accurate, the depth segment is encoded with a standard H.264/AVC Intra codec. The selected coding mode is signalled in the bit stream with one bit per segment. At the decoder side the texture image is reconstructed and segmented; then each segment in the depth map is reconstructed from the corresponding planar coefficients or the H.264/AVC Intra bit stream, depending on the specific segment coding mode. Experimental results on multiple test images show that the proposed scheme can outperform H.264/AVC Intra in terms of quality of reconstructed depth data without affecting computational complexity significantly. Moreover, due to the prediction of depth surface shapes from the texture component, the proposed algorithm can preserve sharp edges better than H.264/AVC Intra on average, depending on the quality of the reconstructed texture. The proposed algorithm has also been tested in the multi-view image coding framework presented in **PAPER 8** – with minor changes to ensure compatibility – as an alternative to standard H.264/AVC Intra coding. Results show that the segmentation-based approach can improve the overall performance of the warping-based multi-view image codec.

PAPER 6: Lossless Compression of Stereo Disparity Maps for 3D

This paper presents a lossless coding scheme for stereo disparity maps. The algorithm encompasses three steps: intra-coding of the

left map, prediction of the right map via disparity warping, and inter-coding of the right map. The left disparity map is coded by means of bit-plane decomposition, Gray coding, and context-based arithmetic coding with adaptive template selection. For each bit-plane, template pixels are chosen among a 30-pixel causal search area in a greedy fashion: the pixel in the search area that minimizes the ideal code length required to encode the bit-plane is included in the template. The procedure is iterated among the remaining pixels until any additional pixel does not reduce the ideal code length anymore. The prediction of the right disparity map is done using a simple warping procedure in which pixels are horizontally shifted according to their value. No vertical shift is required as rectified maps are considered. The right map is encoded as the left one, with the difference that template pixels are chosen among a wider search area: for each bit-plane a 25-pixel non-causal search area from the corresponding bit-plane in the predicted map is considered in addition to the 30-pixel "local" search area. In this way, statistics on not-yet-coded portions of the bit-plane are introduced in the context-coding process, with a positive impact on its performance. Experimental results show that the proposed inter coding method can effectively exploit the predicted map: average bit rate savings between 29% and 52% are reported over the proposed intra method, depending on the resolution of the disparity maps. Average compression factors between 18:1 (low resolution maps) and 48:1 (high resolution maps) are obtained for the considered stereo maps. The proposed method outperforms the most popular solutions for lossless image compression, including JPEG-LS [142], JPEG2000 lossless [143], H.264/AVC lossless in intra-only and predictive configurations, CALIC [144], and bit-plane-based JBIG [145]. Moreover, the scheme features a structure suitable for parallel implementations and provides a progressive data representation, with the possibility of decoding only the left (intra) map from a portion of the bit stream.

PAPER 3: Edge-preserving Intra Depth Coding based on Context-coding and H.264/AVC

In this paper a new Intra mode for edge-preserving intra depth

coding based on H.264/AVC is presented. The proposed algorithm targets depth macroblocks with arbitrarily-shaped edges, in which the Discrete Cosine Transform (DCT) usually fails to provide compact representations. As these blocks are of crucial importance for depth-based view synthesis algorithms, the preservation of their edge information is highly desirable. The proposed Intra mode is integrated in the Rate-Distortion (RD) optimization process, i.e. the mode is tested together with the standard Intra modes of H.264/AVC, and the mode providing the lowest RD cost is selected. The algorithm partitions edge macroblocks into two regions each approximated by a flat surface described by a single scalar value. Correspondingly, a 16×16 binary mask identifying the pixels in the two regions is defined. Binary masks are encoded by means of context-based arithmetic coding with adaptive template selection. As a novel contribution, context statistics of previously encoded edge macroblocks can be exploited to improve the context-coding performance. Specifically, the proposed Intra mode can operate with three different modalities. In the first one binary mask and constant values are encoded without using any prediction from neighboring blocks. The second and third modalities are tested only if at least one of the neighboring macroblocks has been encoded with the proposed coding mode. In the second modality the binary mask is encoded as in the first case but the same constant values of the neighboring edge macroblock are used, avoiding spending bits to encode them. The third modality differs from the second one in the encoding of the binary mask: in this case context statistics of the neighboring edge macroblock are used to initialize the arithmetic encoder. In this way the current binary mask is encoded as part of a bigger mask avoiding the reset of the context status. These two modalities are suitable for those blocks that separate between the same foreground and background objects of the previous edge macroblock (as the same constant values are used) and present similar edge direction/structure (third modality) or different (second modality). Experimental results show that the proposed Intra mode can provide major Peak Signal-to-Noise Ratio (PSNR) gains compared with standard H.264/AVC Intra coding. Improvements in terms of depth quality (up to about 6 dB)

and view synthesis performance using reconstructed depth data (up to about 4 dB) are noticed, depending on the specific test sequences.

PAPER 1: Edge-preserving Intra Mode for Efficient Depth Map Coding based on H.264/AVC

This paper presents an extended version of the scheme proposed in **PAPER 3**. The main improvement consists in the fact that the proposed edge-aware Intra mode is enabled in both Intra (I) and Inter (P) slices, while in **PAPER 3** only I slices are addressed. In this way efficient encoding of depth edges can be performed also in predicted frames. The availability of the proposed Intra mode in P slices allows coding performance comparisons with a number of edge-aware depth coding methods proposed in the literature. Comparisons are made according to four different setups: (1) PSNR of the reconstructed depth signal versus depth bit rate, (2) PSNR of synthesized views from uncompressed texture and reconstructed depth maps versus depth bit rate, (3) PSNR of synthesized views from reconstructed texture and depth maps (encoded with the same QPs) versus texture plus depth bit rate, (4) convex hull of RD points defined by PSNR of synthesized views from reconstructed texture and depth maps (independently encoded with a number of QPs) and texture plus depth bit rate. The paper provides performance comparisons against 8 coding methods including algorithms based on Platelet representations [84], multidimensional multiscale parsing [92], intra prediction with linear residue approximation [85], graph-based transforms [87], block-adaptive palette-based prediction [93], edge-aware intra prediction for 4×4 and 16×16 blocks [146], plane segmentation based intra prediction [77], and H.264/AVC. Experimental results show that the proposed method outperforms a standard H.264/AVC codec on all the considered test sequences in all the comparison setups, with major improvements for the sequences *Ballet* and *Breakdancers*, in which depth maps feature sharp and clean edges. Methods based on Platelets and multidimensional multiscale parsing are also outperformed when one single frame is considered, while for the method in [85] comparable performance are noticed at low bit rates; at high

bit rates the proposed method gains up to 1 dB in terms of synthesized view PSNR. Performance improvements are noticed also when comparing with [93] in terms of reconstructed depth signal PSNR, and with [87] in terms of synthesized view PSNR. Finally, a comparison with [146] and [77] is provided: the proposed method shows significantly better performance than the method in [146], and comparable results with the more complex solution in [77].

3.2 Distributed Approaches for Video-plus-Depth Coding

PAPER 5: Wyner-Ziv Coding of Depth Maps Exploiting Color Motion Information

This paper proposes a new approach for distributed coding of depth images exploiting corresponding texture data, based on the state-of-the-art Distributed Video Coding (DVC) decoder presented by Huang *et al.* [136]. Texture motion information is used at the decoder side to improve the performance of the depth Side Information (SI) generation algorithm, under the assumption that texture and depth share the same motion behavior. In a first setup a Group Of Pictures (GOP) of length 2 is considered. When longer GOP structures are selected, a hierarchical coding structure [136] is used. Texture frames at time instants $t-1$, t, and $t+1$ (referred to as T_{t-1}, T_t, and T_{t+1}, respectively) are available at the decoder, together with depth key frames at instants $t-1$ and $t+1$, referred to as D_{t-1} and D_{t+1}, respectively. The depth Wyner-Ziv frame D_t at instant t is usually decoded using motion estimation and compensation techniques between D_{t-1} and D_{t+1}, e.g. Overlapped Block Motion Compensation (OBMC) proposed in [136]. Since T_t is also known, it can be exploited for more accurate motion estimation. The proposed algorithm first calculates the Motion Vector (MV) fields between T_t and T_{t-1}, and between T_t and T_{t+1} using the so-called Adaptive Rood Pattern Search (ARPS) motion estimation algorithm [138]. Then, the two motion fields are used to motion compensate the depth frames D_{t-1} and D_{t+1} obtaining two different SIs. The arithmetic average of the two SIs is used as SI for the

to-be-decoded depth frame D_t. Experiments have been performed on the sequences *Ballet* and *Breakdancers* down-sampled to QCIF and CIF resolutions. GOP lengths of 2, 4, and 8 were considered for both resolutions. RD performance comparisons with the decoder in [136] and DISCOVER [147] (a reference DVC codec used for benchmarking) are reported, both applied to the depth component only. Results show that the proposed method outperforms the reference codecs for any combination of sequence, GOP, and resolution, demonstrating that texture data can be effectively exploited to improve Wyner-Ziv coding of depth video. Bjøntegaard bit rate savings [148] over the decoder in [136] between 24.4% and 29.8% are noticed for Wyner-Ziv frames for the *Ballet* sequence, depending on GOP length and resolution. As for the *Breakdancers* sequence, bit rate savings between 0.5% and 16.6% are reported. In this case improvements are less marked because the sequence features an intense motion activity and significant occlusions between frames. Consequently, the motion estimation procedure introduces a number of wrong matches, which reflect in noticeable artifacts in the decoded depth signal.

PAPER 4: Distributed Multi-Hypothesis Coding of Depth Maps using Texture Motion Information and Optical Flow

In this paper an extended version of the algorithm for distributed coding of depth maps exploiting texture data presented in **PAPER 5** is proposed. Compared with **PAPER 5** in which one SI is generated at the decoder, the proposed method features the generation of three SIs and an efficient SI fusion technique. The first SI is generated by means of motion estimation from depth data using the OBMC algorithm presented in [136]. In this case no texture information is exploited. The second SI is generated by applying block-based texture MVs – calculated with the ARPS algorithm [138] – to depth frames, as done in **PAPER 5**. The third SI is generated in the same manner, but using optical flow techniques [137] for motion estimation. Optical flow algorithms produce dense motion data (i.e. the MV of every single pixel is estimated) and usually provide more accurate descriptions of motion than standard block-based approaches. In this paper an optical flow estimation algo-

rithm based on [149] is used. The three SIs present different characteristics and provide accurate estimation of the to-be-decoded depth frame in different regions. Therefore, a proper combination can provide significant distortion improvements compared with the three single cases. The SIs are fused using a multi-hypothesis decoder [139], which implements a rate-based optimization strategy by using a number of parallel decoders. Multi-hypothesis decoders proved to be robust SI fusion techniques also in the case of mixed block-based and pixel-based motion estimates [139]. While in **PAPER 5** uncompressed texture data are supposed to be available at the decoder side, this paper considers a more realistic scenario in which texture compression is also addressed. In order to evaluate the robustness of the proposed method to texture compression, depth coding performance are compared with different levels of texture quantization. Experimental results show that the SI based on optical flow estimation outperforms the other SIs when a single SI is used for decoding. Moreover, it shows high robustness against texture quantization. The multi-hypothesis decoder provides better RD performance than the best single SI decoder, confirming that a proper combination of the SIs is beneficial in terms of quality of the decoded signal. Average Bjøntegaard bit rate savings between 21% and 49% over a single SI decoder using OBMC on depth maps are reported for Wyner-Ziv frames.

PAPER 2: Texture Side Information Generation for Distributed Coding of Video-Plus-Depth

This paper proposes a new SI generation scheme for texture data exploiting corresponding depth information in a distributed video-plus-depth coding scenario based on the scheme presented in [136]. In the considered scenario a GOP of length 2 is used. The following frames are supposed to be available for the decoding of the texture frame at time instant t: texture frames at instants $t-1$ and $t+1$, and depth frames at instants $t-1$, t, and $t+1$. As in **PAPER 4**, optical flow techniques are used, multiple SIs are generated at the decoder side, and multi-hypothesis decoding [139] is used to efficiently fuse the various SIs. The main novelty introduced in this paper is an optical flow formulation in which the

texture and depth components are jointly addressed: the optical flow is defined through the minimization of a linear combination of two data terms depending on texture and depth, plus an additional term to penalize irregular behaviors. By modifying the weight coefficients of the linear combination three setups are obtained: (1) texture-only motion estimation, (2) depth-only motion estimation, (3) joint depth-texture motion estimation to allow for more precise pixel matching while properly handling illumination changes and shadows. As in **PAPER 4**, motion fields are used to motion compensate the available texture frames and generate SIs for the to-be-decoded texture frame. Specifically, three SIs based on optical flow estimation are obtained: "T2T", "D2T", and "DT2T", corresponding to (1), (2), and (3), respectively. Two other SIs are considered: a block-based version of D2T employing MVs obtained with the ARPS motion estimation algorithm [138], and the standard SI used in [136], based on the OBMC algorithm applied to texture data only, referred to as "OBMC". Experimental simulations have been performed considering different multi-hypothesis decoders using 2 or 3 SIs. Four combinations have been considered for the decoder with 2 SIs, each including the OBMC SI and one of the other presented SIs. The OBMC, T2T, and DT2T SIs have been selected for the decoder with 3 SIs. Coding of depth maps has also been considered: simulations with different levels of quantization of depth images have been carried out to study the robustness of the proposed method to depth compression. Results on the sequences *Dancer*, *Ballet*, and *Breakdancers* show that the proposed multi-hypothesis decoders outperform the reference decoder [136] for any considered depth quantization level. Specifically, the best RD performance are obtained with the decoder using 3 SIs, which provides average Bjøntegaard bit rate savings on Wyner-Ziv frames between 14% and 31% compared with the state-of-the-art single SI decoder in [136]. The decoder using OBMC and DT2T as SIs shows the best average RD performance among the considered decoders with 2 SIs. Experiments also show that the decoders using SIs based on optical flows are very robust to depth quantization noise.

Chapter 4

Conclusion

Three-Dimensional Video (3DV) is expected to be the next major step in the development of motion picture formats. Both the industry and academia are putting great efforts to overcome the challenges that the new technology brings, and significant results have already been achieved. Among the many issues that need to be faced in the implementation of a 3DV communication system, data compression and transmission plays an important role. 3DV representations that enable the decoupling of content creation, transmission and display formats are of particular interest, and the Multi-view Video-plus-Depth (MVD) representation is currently the most widely adopted solution. Compression of depth information is of crucial importance as view synthesis algorithms used at the decoder side are very sensitive to depth accuracy, especially around edge regions. Motivated by the fact that depth images exhibit different structure and characteristics compared with natural images, a number of non-standard solutions for depth map coding have been presented in the literature, as broadly discussed in Sec. 1.2.4. The general goal is to develop coding schemes able to provide good compression gains while preserving sharp discontinuities even at low bit rates.

This Ph.D. project focused on new approaches for efficient coding of depth information. A hybrid solution based on H.264/AVC Intra, surface shape prediction via texture segmentation, and planar fitting was proposed, with interesting results in terms of compression gain – up to about 4 dB in terms of Peak Signal-to-Noise Ratio (PSNR) compared with H.264/AVC Intra – and edge preservation. Standard H.264/AVC

coding of depth maps was also studied for efficient warping-based multi-view image coding, showing PSNR improvements up to 1.2 dB compared with a standard H.264 MVC codec. An edge-aware Intra mode based on H.264/AVC was also presented. Depth macroblocks featuring arbitrarily-shaped edges are approximated by two flat regions separated by a sharp edge. Edge information is conveniently encoded by means of context-based arithmetic coding exploiting previously-encoded edge blocks, allowing for significant Rate-Distortion (RD) cost reductions compared with the standard Intra modes. For the case of depth maps featuring sharp and clean edges, the proposed mode gives substantial RD improvements in terms of Bjøntegaard bit rate savings (with reductions up to 65%) and view synthesis accuracy. Most of the presented schemes can be adapted to the upcoming H.265/HEVC standard with minor changes, making them suitable to encode depth maps for content in High Definition (HD) and beyond. Lossless coding of depth maps was addressed too, with an efficient scheme for stereoscopic disparity maps based on depth warping and context-based arithmetic coding with adaptive template selection, able to provide bit rate reductions between 22% and 76% compared with standard lossless codecs. Finally, novel approaches for distributed video-plus-depth coding were proposed. When the motion correlation between depth and texture is conveniently exploited for Side Information (SI) generation purposes e.g. by using joint depth-texture optical flow calculations, Bjøntegaard bit rate savings up to 49% compared with state-of-the-art schemes can be obtained on Wyner-Ziv frames. The proposed schemes can be used also in classical (non-distributed) video-plus-depth coding scenarios as a way to reduce inter-component redundancy e.g. at the encoder side. Extensions from mono-view to multi-view formats can be derived too.

Future Work and Discussion

Coding approaches exploiting graph-based lifting transforms, soft decoding, and inpainting techniques based on Partial Differential Equations (PDEs) have also been explored. Unfortunately, the considered setups did not provide competitive performance. The further development of these schemes – which are still believed to be potentially efficient when properly optimized – is left as future work. A coding scheme featuring explicit edge coding and PDE-based inpainting has been presented

by Gautier *et al.* [150] recently, confirming that the approach fits with the peculiar properties of depth images.

As for future research directions in the field of 3DV coding, explicit geometric data are expected to remain included in future representations, and (layered) depth maps will be the most common way of representing geometric information. Coding strategies exploiting inter-component correlation – especially in terms of motion and object shapes – will ensure high compression performances, as done in the 3D extension of H.265/HEVC, currently under development [151]. Or perhaps a combination of some of the over 200 depth coding methods in the literature will provide the best tradeoff between compression performance, robustness to inaccuracies, and computational complexity. As new display technologies appear and consolidate, the need for more sophisticated representations may rise in the long term, requiring more flexibility in terms of rendering capabilities. As an example, vertical parallax is not handled by current display technologies in the market, limiting the possibilities of free-viewpoint rendering. Future representations may specifically address this and other issues (e.g. the currently problematic acquisition of accurate depth data), maybe exploiting hybrid solutions in which object-based and image-based representations are conveniently combined. The increasing popularity of high-performing mobile devices such as smartphones and tablets may also affect the evolution of 3DV coding and distribution technologies, favoring solutions that enable high robustness to data loss and efficient real-time communication.

Appendices

Appendix A

Ph.D. Publications

Edge-preserving Intra Mode for Efficient Depth Map Coding based on H.264/AVC

M. Zamarin, S. Forchhammer*

Technical University of Denmark, Department of Photonics Engineering, Ørsteds Plads B.343, 2800 Kongens Lyngby, Denmark

Abstract

Depth-Image-Based-Rendering (DIBR) algorithms for 3D video communication systems based on the "Multi-view Video plus Depth" format are very sensitive to the accuracy of depth information. Specifically, edge regions in the depth data should be preserved in the coding/decoding process to ensure good view synthesis performance, which directly affects the overall system performance. This paper proposes a novel scheme for edge-aware intra depth compression based on the H.264/AVC framework enabled on both intra (I) and inter (P) slices. The proposed scheme includes a new Intra mode specifically targeted to depth macroblocks with arbitrarily-shaped edges, which are typically not predicted well by the standard Intra modes of H.264/AVC and result in high rate-distortion costs. The proposed algorithm segments edge macroblocks into two regions each approximated by a flat surface. A binary mask identifying the two regions is defined and encoded by means of context-coding with adaptive template selection. As a novel contribution, the proposed mode allows exploiting the correlation with causal neighboring edge macroblocks to improve the performance of context-coding of binary masks and allow significant bit rate savings. The proposed method has been exhaustively compared with different state-of-the-art algorithms for edge-aware depth coding and the results highlight significant improvements in most of the cases, both in terms of reconstructed depth quality, view synthesis performance, and overall texture plus depth rate-distortion performance.

Keywords: multi-view plus depth video coding, edge-aware depth map coding, context based coding, depth-image-based rendering, H.264/AVC
2010 MSC: 68P30, 94A08

1. Introduction

3D video communication technologies are becoming more and more popular nowadays. The most promising 3D video representation is the so-called "Multi-view Video

*Corresponding author. Tel.: +45 45253622
Email addresses: mzam@fotonik.dtu.dk (M. Zamarin), sofo@fotonik.dtu.dk (S. Forchhammer)

M. Zamarin, S. Forchhammer, "Edge-preserving Intra Mode for Efficient Depth Map Coding based on H.264/AVC", *Elsevier Signal Processing: Image Communication (submitted).*

plus Depth" (MVD) in which together with multi-view data also depth images are provided. The availability of depth data allows synthesizing the desired output views at the decoder side to fit the requirements of the specific display device. In this manner only a limited number of views – typically 2 or 3 – needs to be transmitted without compromising the rendering capabilities at the decoder. Coding frameworks based on the MVD format are currently being developed and standardized by the ISO/IEC JTC 1/SC 29/WG 11 (MPEG) 3DV ad-hoc group [1], which recently joined the efforts with the ITU-T SG 16 WP 3 to form the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V) [2]. Solutions compatible with both the current H.264/AVC [3] and the upcoming H.265/HEVC (High Efficiency Video Coding) [4] standards are being investigated.

View synthesis algorithms based on the Depth-Image-Based-Rendering (DIBR) paradigm [5] are currently used to generate virtual views at the decoder side from reconstructed texture and depth signals. These algorithms are highly sensible to depth inaccuracies [6], especially around sharp edges. Therefore, steep discontinuities in the depth signal should be preserved in order to allow the rendering of high-quality virtual views, which directly affect the overall 3D video system performance. A number of depth coding algorithm have been proposed in the literature to cope with this issue, for instance [7] in which edge-adaptive transforms are defined and depth values are manipulated to improve the compression performance, or [8] in which depth edge blocks are segmented and each segment approximated with a planar surface. A discussion of recent works in the field of edge-aware depth map coding is provided in Sec. 2.

This paper presents a novel algorithm for efficient edge-preserving depth map coding extending the preliminary version presented in [9]. The proposed algorithm is based on the H.264/AVC framework and defines a new Intra mode for full MacroBlocks (MBs), tested together with the Intra modes defined in the standard in both Intra (I) and Inter (P) frames (as one of a number of extensions of [9] in which only I frames are addressed). The proposed mode is specifically targeted to arbitrarily-shaped edge MBs, in which DCT typically fails to provide compact representations. MBs are partitioned into two regions according to the edge structure, and each region is approximated with a constant value. Regions are identified through a binary mask that is losslessly encoded by means of context-coding with adaptive template selection. As a novel element, the proposed mode is able to exploit previously encoded edge MBs to predict the two constant values and the edge structure of the current MB to further improve the compression gain.

The remainder of this paper is organized as follows. Section 2 gives an overview of related works in the field of edge-aware depth coding. Section 3 presents the proposed edge-aware Intra mode, Sec. 4 provides an extensive performance comparison of the proposed method against a number of recent depth coding methods highlighing the benefits of the proposed algorithm.

2. Related Work on Edge-aware Depth Map Coding

Different algorithms for efficient coding of depth maps for 3D video communication systems based on the MVD format have been proposed in the last years in the literature. The general goal is to represent sharp edges in efficient manners to allow high-quality view synthesis at the decoder side. The most straightforward approach for depth map coding is the use of standard video codecs such as H.264/AVC [3]. If on one hand this

solution allows taking advantage of all the expertise developed for video coding, on the other hand it does not provide an optimal solution due to the substantial structural differences between natural images and depth maps. Depth maps feature wide flat or smooth areas divided by sharp edges. While smooth areas are very well represented by DCT (i.e. very few non-zero transform coefficients are produced), edge blocks typically require many coefficients and are very expensive in terms of Rate-Distortion (RD) cost. For this reason, alternative solutions – based both on standard approaches and completely new frameworks – have been proposed. Depth coding algorithms can be grouped into two categories: those that exploit the correlation with the corresponding texture signal, and those that do not rely on inter-component correspondencies.

2.1. Coding of depth data without exploiting corresponding texture

Morvan *et al.* [10, 11] developed a Platelet-based coding algorithm in which non-overlapping depth blocks are defined using a quadtree decomposition strategy. Within each block the depth signal is approximated using piecewise-linear functions separated by a straight line (platelets). The method was later combined with H.264/AVC-like intra prediction and a dictionary-based approach by Lucas et al. [12] to improve the compression performance. Graziosi *et al.* [13] proposed a pattern matching-based encoder called "Multidimensional Multiscale Parser" able to preserve and efficiently encode high frequency patterns, differently from the traditional DCT-based approaches.

Shimizu *et al.* [14] developed a coding algorithm based on a similar idea as the one proposed in this paper: edge MBs are approximated with a binary palette and an object shape map predictively encoded exploiting both intra- and inter-frame correlations. Shape maps are generated with the minimization of a prediction error function and encoded by means of context-adaptive binary arithmetic coding similarly to shape coding in MPEG-4 Part 2. Differently from the proposed method, the algorithm in [14] does not explicitly exploit the edge structure of previously encoded edge MBs, which can be of benefit especially at low bit rates.

As in the proposed method, Oh *et al.* [8] also proposed adding one new Intra mode specifically targeted to edge blocks to a standard H.264/AVC encoder. Blocks are segmented in a number of regions depending on the edge structure and a planar approximation is calculated for each segment. Segmentation information is encoded in an efficient manner by means of CABAC. The method is implemented on 4×4, 8×8 and 16×16 blocks. Experimental results show significant gains over a standard H.264 MVC coder and over a previously proposed edge-aware intra prediction method [15] in which constant approximation was performed instead of planar on 4×4 blocks. However, as in the previous case, the edge structure of previously-encoded blocks is not directly exploited during the encoding of segmentation information.

Lan *et al.* [16] based their solution on similar concepts for the case of HEVC. Block sizes ranging between 4×4 and 32×32 are considered. Blocks are segmented in up to 8 arbitrarily-shaped regions each approximated by a flat surface. Partitioning information is encoded by means of arithmetic coding. The algorithm shows significant improvements over the reference HEVC for both acquired and generated depth maps.

Kim *et al.* [17] proposed to use graph-based transforms as an alternative to standard DCT. The transform addresses 4×4 edge blocks in an efficient manner, but increases the overall complexity because of the explicit calculation of the graph Laplacian for each block.

2.2. Coding of depth data exploiting corresponding texture

Cheung *et al.* [18, 19] proposed to manipulate depth values in order to sparsify the input signal to make it more suitable for compression. Depth pixels are manipulated according to their influence in the synthesis process: the less influent, the more they can be modified if beneficial in terms of depth compression gain. The algorithm was then combined with the edge-adaptive transforms presented in [20] (an early version of [17]) to further improve the performance [7].

Merkle *et al.* [21] proposed to exploit edges in the co-located reconstructed texture data to segment depth blocks into two regions, each approximated by a flat surface. Since the same block partitioning can be reproduced at the decoder due to the availability of the texture, only the flat surface values need to be transmitted. The algorithm, integrated in the HEVC framework, shows promising coding gains but performances strongly depend on the alignment of edges between texture and depth, which may change significantly from sequence to sequence.

Texture edge information is also exploited in the method proposed by Milani *et al.* [22] in which reconstructed texture data are segmented and segments are used to predict shapes in the corresponding depth maps. Within each segment, depth values are approximated with a planar surface, whose coefficients are encoded and transmitted. Due to the fact that segmentation data can be reproduced at the decoder from the reconstructed texture, depth edges are not explicitly encoded allowing significant bit rate savings, depending on the quality of the decoded texture. A framework based on a similar concept has been proposed by Ruiz-Hidalgo *et al.* [23] for the case of multi-view depth video. Texture segmentation information is refined by additional data in order to fit accurately shapes in the depth signal. Within each segment, depth values are encoded exploiting a decomposition into orthogonal basis functions. Significant performance improvements are noticed compared with standard H.264 MVC coding.

3. Proposed Edge-preserving Intra Mode

The proposed Intra mode – called "EDGE" mode in the following – operates at a MB level as part of the RD optimization stragegy, i.e. for each MB the RD cost provided by the proposed mode is compared with the costs of the other modes and the mode resulting in the minimum cost is selected. The proposed mode can operate in three different ways, namely *EDGE Intra*, *EDGE Inter CV*, and *EDGE Inter full*. In all cases no temporal redundancies are exploited: the name *Inter* refers to the fact that previously encoded EDGE MBs in the same slice are exploited during the coding process, differently from the *EDGE Intra* case in which no causal data are exploited. The three modalities are now discussed in detail.

3.1. EDGE Intra mode

This mode operates without exploiting previously encoded blocks. As mentioned in Sec. 1, the proposed mode partitions an edge MB into two regions and approximates each of them with a flat surface described by a Constant Value (CV). A 16×16 binary mask M defining the two regions is losslessly encoded by means of context-based arithmetic coding, together with the two CVs. Details of the *EDGE Intra* mode are reported below and a block scheme is shown in Fig. 2(a).

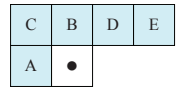


Figure 1: Candidate pixels for the template. Pixels A and B are fixed, a third pixel among C, D, and E can be included in the template. The current pixel being encoded is indicated by "•".

1. The input MB is partitioned into two regions using a simple clustering algorithm. Using pixel coordinates and depth values, a 3D-point is associated to each pixel in the MB. Each 3D-point is connected with its closest n neighbors (considering Euclidean distances), the initial value of n being 1. If more than two components are obtained, n is increased and the procedure iterated until two groups are obtained. A 16×16 binary mask M identifying the two components is defined.
2. A CV is associated to each component of M. The CV of a component is defined by the median value of the corresponding pixels in the input MB.
3. The binary mask M is encoded by means of context-based arithmetic coding. The template is defined among a small set of 5 candidate pixels, reported in Fig. 1. Pixels A and B are always included in the template. A third pixel among C, D, and E can be included if of benefit for the code length. The mask M is encoded with the different template configurations and the template requiring the minimum number of bits R_M is selected. Let R_t be the number of bits spent to specify the size of the template and the third template pixel, if used, and R_{CV} the number of bits to losslessly encode the two CVs. The edge statistics (i.e. the final status of the arithmetic encoder) are stored for the current MB, if encoded as EDGE MB, and made available for the next MBs being encoded. Note that these statistics do not need to be encoded as they can be generated at the decoder side during the decoding process.
4. The distortion D between the input depth MB and the binary EDGE MB is calculated as Mean-Squared-Error (MSE).
5. The RD cost of the *EDGE Intra* mode is calculated as for the standard Intra modes of H.264/AVC: $RD^{intra}_{EDGE} = D + \lambda R$, where $R = R_M + R_t + R_{CV}$.

Experimental tests showed that the clustering procedure described in Step 1 is on average more accurate than simpler bipartition methods like for instance the one used in [24] in which only depth values are considered. Experiments showed that very few iterations are typically enough to converge to two components. The choice of the median value over the mean value in Step 2 is motivated by experimantal tests: the median value appears to be on average more robust to outliers (i.e. noisy pixels in the MB partition) and has been therefore chosen.

3.2. EDGE Inter CV mode

The *EDGE Inter CV* mode is tested if at least one of the left-neighboring and top-neighboring MBs has been encoded as an EDGE MB. In this case, the same procedure described for the *EDGE Intra* mode is followed, with the difference that the two CVs are not calculated and encoded but copied from the "predicting" EDGE MB (i.e. the CVs

of the previously encoded EDGE MB are used as predictors for the current CVs and no residuals are encoded). In this way no bits are spent to encode the CVs, allowing for bit rate savings in the cases in which edges spanning adjacent MBs separate between the same background and foreground objects. If the left- (top-) neighboring MB is encoded as EDGE MB, the cost RD^{left1}_{EDGE} $\left(RD^{top1}_{EDGE}\right)$ is defined. Figure 2(b) shows a block scheme of the proposed *EDGE Inter CV* mode.

3.3. EDGE Inter full mode

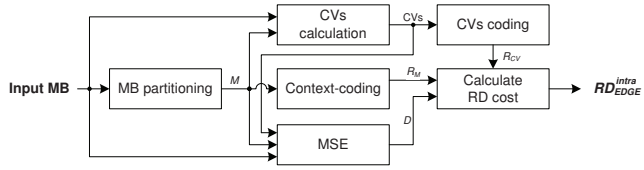
The *EDGE Inter full* mode represents the main novelty of the proposed method. In this mode not only the CVs are predicted from a neighboring EDGE MB as in the *EDGE Inter CV* mode, but also the template pixels used for the context-coding. Since the same template will be used, the edge statistics of the predicting EDGE MB can be used as a starting point for the context-based arithmetic coding. In this way the binary mask of the current MB is encoded as part of a bigger binary mask spanning one or possibly more adjacent EDGE MBs. For the case of long edges with a slow-varying structure/direction this mode allows major bit rate savings as no bits are spent for the CVs and the arithmetic encoder exploits the statistics of the already encoded portion of the edge. As for the proposed *EDGE Intra* and *EDGE Inter CV* modes, when the *EDGE Inter full* mode is selected the edge statistics are stored thus allowing the following edge MBs to be encoded as part of the same edge region. If the left- (top-) neighboring MB is encoded as EDGE MB, the cost RD^{left2}_{EDGE} $\left(RD^{top2}_{EDGE}\right)$ is defined. The *EDGE Inter full* mode block scheme is summarized in Fig. 2(c).

The minimum RD cost among RD^{intra}_{EDGE} and RD^{left1}_{EDGE}, RD^{top1}_{EDGE}, RD^{left2}_{EDGE}, RD^{top2}_{EDGE} (if defined) is used to select the best EDGE mode to be compared with the standard coding modes of H.264/AVC.

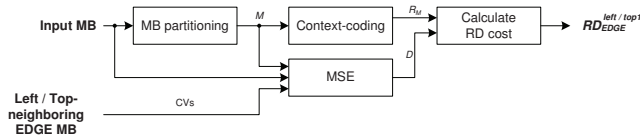
The availability of the two *EDGE Inter* modes allows for efficient handling of edges developing over multiple MBs. The *EDGE Inter full* mode is particularly suited to long edges that exhibit a regular structure/direction (so that the same template remains optimal among different MBs), while the *EDGE Inter CV* mode captures changes in the edge structures. In case the depth values in the regions separated by the edge change significantly from MB to MB, the *EDGE Intra* mode is more likely to be selected.

The different EDGE modes are signalled with few bits encoded by means of CABAC. When Intra 16×16 modes are tested, one bit is encoded to specify whether standard or EDGE modes are used. If an EDGE mode is used, one extra bit is encoded in the case that at least one of the left- or top-neighboring MBs has been encoded as EDGE MB to signal whether *EDGE Intra* or *EDGE Inter* coding is used. If *EDGE Inter* coding is used, one additional bit signals which MB is used for prediction (if both left- and top-neighboring MBs have been coded as EDGE MBs) and a last bit indicates which of the two *EDGE Inter* modes is selected.

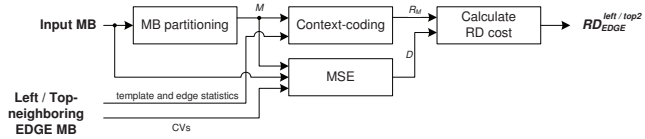
At the decoder side, if an EDGE MB is detected the binary mask is decoded by means of context-based decoding, the CVs are decoded or copied from a selected neighboring EDGE MB, depending on the EDGE mode, and the output EDGE MB is produced. In case the EDGE mode is not used, the MB is decoded following the standard Intra decoding procedures.



(a) Proposed *EDGE Intra* mode



(b) Proposed *EDGE Inter CV* mode



(c) Proposed *EDGE Inter full* mode

Figure 2: Block schemes of the three proposed EDGE modalities.

4. Experimental Results

The presented EDGE mode has been implemented on the H.264/AVC reference software JM version 17.1. A number of 3D video test sequences with different resolutions and characteristics have been considered for the validation of the proposed method, namely *Ballet* and *Breakdancers* [25], *Balloons*, *BookArrival*, *Cg*, *ChampagneTower*, *Interview*, *Kendo*, *Lovebird1*, *Mobile*, *Orbi*, *Newspaper*.

The proposed scheme has been compared with a standard H.264/AVC coder to validate its effectiveness. As reported in Sec. 2, many edge-aware depth coding algorithms have been proposed recently in the literature. In order to allow for a comprehensive evaluation of the coding performance, the proposed algorithm is also compared with the Platelet-based method [10] (referred to as "Platelets"), the method based on the multidimensional multiscale parser [13] (referred to as "MMP"), the algorithm exploiting linear residue approximation and intra prediction on adaptive blocks [12] (referred to as "LRA"), the method exploiting graph-based transforms [17] (referred to as "GBT"), the algorithm using block-adaptive palette-based prediction [14] (referred to as "BAPP"), the method exploiting edge-aware intra prediction for 4×4 and 16×16 blocks [15] (referred to as "EAIP"), and the method exploiting plane segmentation based intra prediction [8] (referred to as "PSIP"). Due to different test conditions and sequences, a direct comparison with all the mentioned algorithms is not possible. Instead, methods are grouped

and compared according to the specific test conditions.

The coding performance of the various algorithms are evaluated according to four different setups for comparison:

1. **S1. Depth PSNR** - A mono-/stereo-scopic depth video is encoded and decoded, and the PSNR between uncompressed and decoded signals evaluated. RD curves describing depth PSNR against depth bit rate are compared. This test method evaluates the quality of the decoded depth signal only and is suitable for any applications in which depth maps are compressed with lossy algorithms.
2. **S2. Synthesized view PSNR versus depth bit rate** - The depth videos of the left and right views are encoded/decoded. Decoded depth maps and uncompressed views are used to synthesize intermediate views. These views are compared with intermediate views obtained from uncompressed depth maps and uncompressed views. RD curves describing synthesized view PSNR against depth bit rate are compared. This test method allows decoupling the effects of depth compression and those of video compression as texture is only used for view synthesis.
3. **S3. Synthesized view PSNR versus total bit rate** - Left and right views and depth maps are compressed using same QPs and GOP structures. Intermediate views are generated from decoded views and decoded depth maps (encoded with same QPs) and compared with intermediate views generated with uncompressed views and uncompressed depth maps. RD curves describing synthesized view PSNR against total (view plus depth) bit rate are compared. This method allows describing the improvements (i.e. rate or PSNR variation) in the overall video-plus-depth system.
4. **S4. Convex hull** - Left and right views and depth maps are compressed using same GOP structures and all the QPs in a predefined interval. Intermediate views are generated for any combinations of decoded views and depth maps, and compared with intermediate views generated with uncompressed views and uncompressed depth maps. For each combination of decoded views and decoded depth maps a RD point is obtained (the rate includes both views and depth maps rates). The convex hull of the entire set of RD points is calculated. This is an off-line optimization method that shows the best performance achievable in terms of synthesized views PSNR when a flexible bit rate allocation between views and depth maps is allowed.

Synthesized views (S2, S3, and S4) are generated using the View Synthesis Reference Software (VSRS) 3.5 [26]. For each comparison setup, the Bjøntegaard bit rate saving [27] between the considered algorithms and the reference coder can be evaluated and compared.

4.1. Comparison with H.264/AVC

First and foremost the proposed method is compared with a standard H.264/AVC coder. All the four comparison setups have been considered for the sequences *Ballet*, *Breakdancers*, *Kendo*, *Lovebird1*, *Newspaper*. Test conditions and settings are reported in Tables 1 and 2. Figures 3 and 4 report the performance comparisons for the *Ballet* and *Breakdancers* sequences, highlighting the benefits of the proposed method. Performance improvements are summarized in Table 3 for all the considered sequences.

As it can be noticed, the proposed method allows improving the performance of H.264/AVC for all the considered setups for comparison. Major gains are noticed for

Table 1: Test conditions - Performance comparison against standard H.264/AVC

Feature, tool or setting	**H.264/AVC and Proposed**
Number of frames	100
Profile	FRExt High
GOP structure	IPPP
GOP length	8
R-D opt. (incl. Intra 16×16)	Yes
Deblocking filter	No
Entropy coder	CABAC
Block size	Adaptive
Search range	±32
ME accuracy	Full-pel
Depth QPs	24, 28, 32, 36
Texture QPs	24, 28, 32, 36
Depth QPs (S4 only)	24, 28, 32, . . . , 48
Texture QPs (S4 only)	24, 28, 32, . . . , 48

Table 2: Selected views - Performance comparison against H.264/AVC

Sequence	**Views (left, virtual, right)**
Ballet	5, 4, 3
Breakdancers	5, 4, 3
Kendo	3, 4, 5
Lovebird1	6, 7, 8
Newspaper	4, 5, 6

the *Ballet* and *Breakdancers* sequences while minor improvements are noticed for the other sequences. The reason for this difference resides in the fact that the two former sequences have sharp edges, which are encoded very efficiently by the proposed EDGE mode. Maximum PSNR improvements of about 2.0 dB and 1.3 dB are noticed when the overall bit rate is considered (S4) for the *Ballet* and *Breakdancers* sequences, respectively. The sequence *Kendo* only shows a small variation of the total bit rate (S4) due to the fact that the original depth maps contain "false" edges, i.e. edges that do not correspond to edges in the video. For those edges the proposed EDGE mode may give some benefits in terms of depth PSNR (S1), but view synthesis performance are not affected in the same manner.

As it can be seen from the convex hulls (S4) in Fig. 3 and Fig. 4, choosing of the same QPs for texture and depth is not always optimal, both for H.264/AVC and the proposed method: higher overall RD performance can be achieved if depth and texture bit rates are partitioned differently (labels next to the RD points in Figs. 3 and 4 S4 indicate the corresponding texture and depth QPs). Optimal joint bit rate allocation is out of the scope of this paper and has been already addressed in the literature (see for instance [28] and [29]); convex hulls are only used to evaluate and compare the best performance achievable by the methods being considered.

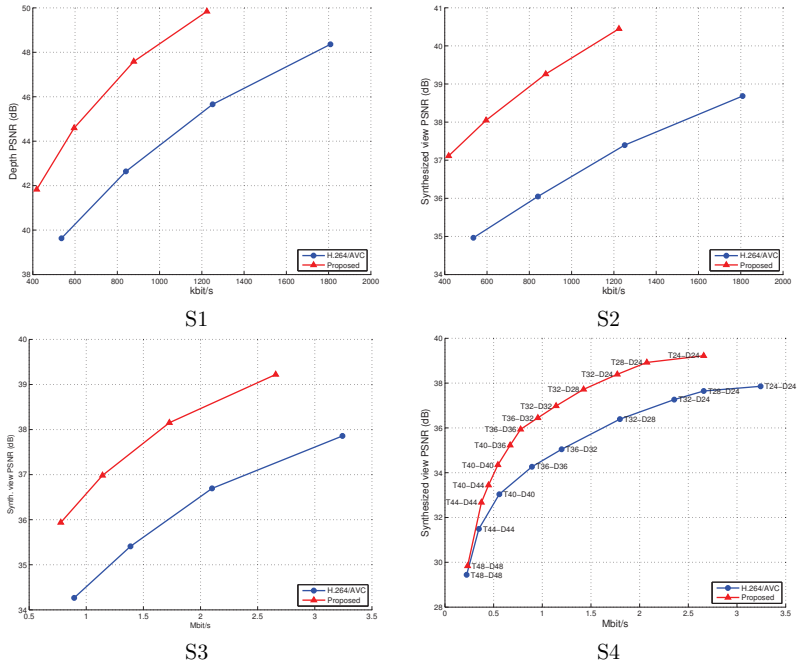


Figure 3: RD performance comparisons between H.264/AVC and the proposed method for the *Ballet* sequence.

Experimental results over all the test sequences show that the average number of bits spent to encode an EDGE MB is about 62 for an *EDGE Intra* MB and about 32 for an *EDGE Inter* MB, showing that the proposed *Inter* modes achieve a significant bit rate reduction of about 50% over the *Intra* mode. The complexities of the proposed and standard Intra modes do not differ significantly as no computationally expensive operations are introduced. Therefore, only a minor increase of the overall system complexity is noticed.

4.2. Comparison with Platelets, MMP, LRA

The performance of the proposed method are compared with those of the Platelet-based algorithm [10], MMP [13], LRA [12], and H.264/AVC on the first frame of the *Ballet* and *Breakdancers* sequences. Comparisons with the Platelet-based method and LRA are of particular interest as both of these methods aim at preserving sharp edges in the depth signal. Differently from the proposed method, these algorithms locally approximate edges with straight lines on variable-size blocks and use planar fitting functions for the non-edge depth regions (see Sec. 2 for more details).

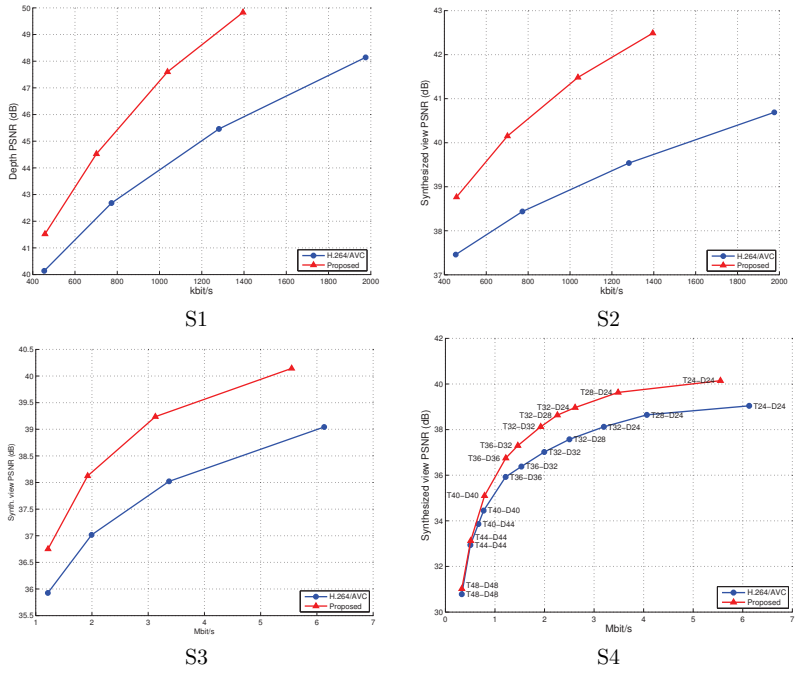


Figure 4: RD performance comparisons between H.264/AVC and the proposed method for the *Breakdancers* sequence.

The S2 setup for comparison is used. The following views have been selecetd: 5 (left), 4 (virtual), and 3 (right) for both sequences. Except for number of frames, GOP length and QPs, the settings are the same as in Table 1. QPs in the set {24, 28, 32, . . . , 48} have been used. As only one frame is encoded, bit-per-pixel values are reported rather than bit-per-second. Figure 5 shows a comparison of the RD performance of all the considered methods.

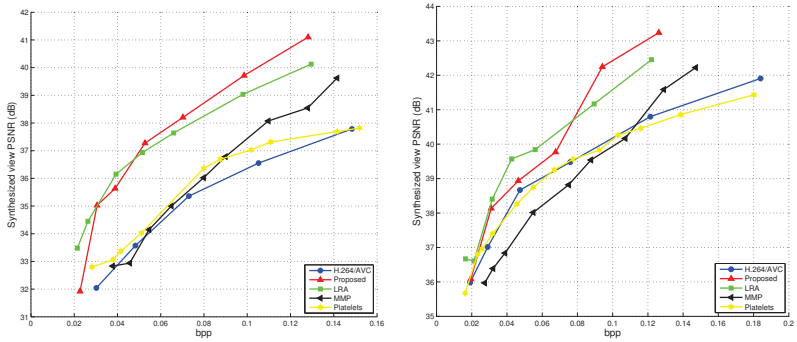
The RD plots show that the proposed method is able to outperform H.264/AVC, MMP, and the Platelet-based algorithm at all the bit rates for both *Ballet* and *Breakdancers* sequences. The comparison with LRA is also satisfactory: at low rates LRA and the proposed method have comparable performance, with some gain of LRA especially in the *Breakdancers* sequence, and at high rates the proposed method outperforms LRA by up to 1 dB in both sequences.

4.3. Comparison with BAPP

A comparison based on the S1 setup has been done against the BAPP method [14] for a number of standard resolution test sequences. The settings in Table 1 were used, except for the GOP length (set to 16) and QPs (set to 22, 27, 32, 37). Results are summarized

Table 3: BDBR savings of the proposed method over H.264/AVC (in %)

Sequence	**S1**	**S2**	**S3**	**S4**
Ballet	−45.04	−64.95	−54.51	−43.05
Breakdancers	−35.87	−53.55	−44.29	−34.33
Kendo	−4.35	−6.81	−1.25	−0.36
Lovebird1	−15.42	−23.69	−3.96	−2.70
Newspaper	−7.93	−8.01	−1.83	−1.45
Average	**−21.72**	**−31.40**	**−21.17**	**−16.38**



a) *Ballet* b) *Breakdancers*

Figure 5: RD performance comparison: proposed method vs. Platelets, MMP, and LRA.

in Table 4 in terms of Bjøntegaard bit rate savings of the BAPP and proposed methods over a standard H.264/AVC coder.

Results show that the proposed method can provide a higher average bit rate reduction over H.264/AVC compared with BAPP. Compared with BAPP, the proposed method achieves an average bit rate saving that is about 14% bigger. The proposed method outperforms BAPP in 7 out of 9 sequences, with the biggest relative improvement on the *Cg* sequence (about 66% more than BAPP), a computer generated sequence featuring sharp depth maps.

4.4. Comparison with GBT

A comparison with GBT [17] has been done following the comparison setup S2. 15 frames have been encoded with a GOP size of 15 (other settings as in Table 1). Table 5 reports the comparison results in terms of Bjøntegaard bit rate savings of GBT and the proposed method over H.264/AVC for six test sequences.

Experimental results highlight that the proposed method can achieve better bit rate reductions in 4 out of 6 sequences compared with GBT, with very similar reductions on the 2 remaning sequences. Overall, a bit rate reduction about 65% bigger than the one provided by GBT is noticed.

Table 4: BDBR savings of BAPP and the proposed method over H.264/AVC (in %)

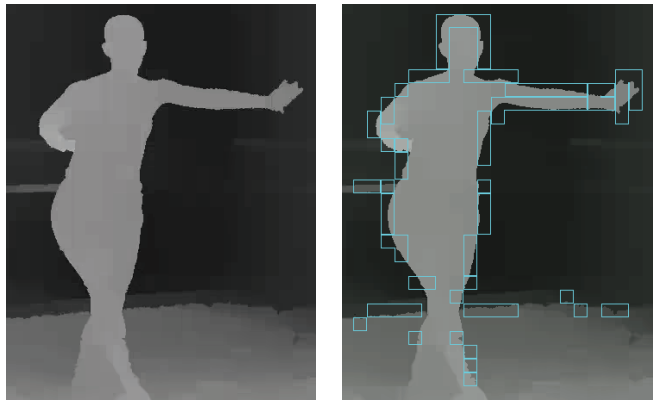
Sequence	**BAPP**	**Proposed**
Ballet	−40.43	**−45.65**
Balloons	**−6.76**	−5.15
BookArrival	**−16.52**	−9.67
Breakdancers	−28.61	**−37.02**
Cg	−13.77	**−22.80**
ChampagneTower	−21.74	**−24.80**
Interview	−24.04	**−26.64**
Lovebird1	−7.15	**−7.21**
Orbi	−5.99	**−8.42**
Average	−18.33	**−20.82**

Table 5: BDBR savings of GBT and the proposed method over H.264/AVC (in %)

Sequence	**GBT**	**Proposed**
Balloons	**−7.1**	−6.7
BookArrival	−17.1	**−38.1**
ChampagneTower	−9.8	**−22.1**
Kendo	−3.9	**−8.0**
Mobile	−24.0	**−31.6**
Newspaper	**−6.4**	−6.3
Average	−11.4	**−18.8**

4.5. Comparison with EAIP and PSIP

Finally, a performance comparison is made with EAIP [15] and PSIP [8] following the results in [8] for the setups S3 and S4. A direct comparison with EAIP and PSIP is not possible as different setups are used: EAIP and PSIP are based on the H.264 MVC reference software JMVC 8.5 while the proposed method is implemented on the H.264/AVC reference software JM 17.1. The proposed method can be seen as a low-complexity version of an MVC-based system (such as PSIP), in which inter-view prediction and hierarchical B pictures have been disabled. Moreover, the proposed EDGE mode is applied to full MBs only, differently from PSIP in which the plane segmentation based intra mode is applied to 4×4, 8×8, and 16×16 blocks. With these differences in mind, the proposed method has been compared with PSIS and EAIP. Comparison results are satisfactory: the proposed method achieves an average Bjøntegaard bit rate reduction over a standard H.264/AVC coder that is slightly higher than the average reduction of PSIP over H.264 MVC when considering the S3 comparison setup on the same test sequences (-16.29% against -15.80%), and a slightly lower average reduction when considering the S4 setup (-5.22% against -5.90%), which makes the proposed method a promising low-complexity alternative to PSIP. Compared with EAIP, the proposed method achieves significantly better results both in terms of S3 (EAIP marks an average bit rate variation of -9.62%) and S4 (-1.88% for the EAIP method).



(a) (b)

Figure 6: Comparison of reconstructed depth maps: (a) H.264/AVC (42.65 dB @ 418 kbit/s)and (b) proposed (44.54 dB @ 298 kbit/s), *Ballet* sequence, detail of frame 4, view 5, both QP 32. The squares in (b) highlight MBs that have been encoded with the EDGE mode. Connected squares indicate the usage of the proposed *EDGE Inter* modes in the corresponding MBs.

4.6. Visual comparison

Figure 6 shows a comparison of a reconstructed depth map encoded with both H.264/AVC and the proposed method. As it can be noticed, the proposed method provides a significantly higher PSNR value and a reduced depth bit rate. Sharp edges are preserved well when the proposed method is selected, confirming that DCT-based coding is not efficient for the case of edge depth MBs. Light-blue squares indicate MBs that have been encoded with the proposed EDGE mode. EDGE MBs encoded using EDGE prediction are represented with squares connected to the predicting EDGE MB. As highlighted, edges spanning adjacent MBs are often encoded with the proposed *EDGE Inter* modes (see for instance the head and the left arm of the dancer), which can represent very efficiently arbitrarily shaped long edges.

In Fig. 7, a visual comparison of synthesized views obtained with depth maps encoded with H.264/AVC and the proposed method is provided. Views have been chosen so that the overall texture plus depth bit rates are similar for the two coding methods. The comparison highlights the advantages introduced by the proposed method: much less artifacts appear in the edge regions providing a more natural appearance of the content. Notice that the texture in Fig. 7(b) also includes more details than the one in (a) as the saving of depth bit rate provided by the proposed method allowed for an increase of the texture bit rate to compare with H.264/AVC at a similar total bit rate. The two images shown in Fig. 7(a) and (b) correspond to the points "T36-D32" (H.264/AVC) and "T32-D32" (proposed) in Fig. 3 S4.

(a) (b)

Figure 7: Visual comparison of synthesized views versus total bit rate (depth plus texture): detail of the *Ballet* sequence, frame 4: (a) H.264/AVC (35.05 dB @ 1.199 Mbit/s), (b) proposed (36.98 dB @ 1.142 Mbit/s). These synthesized views correspond to the points "T36-D32" (H.264/AVC) and "T32-D32" (proposed) in Fig. 3 S4.

5. Conclusion

This paper presents a novel Intra coding mode specifically targeted to depth macroblocks with arbitrarily-shaped edges. The proposed algorithm segments edge macroblocks into two regions and approximates each of them with a flat surface. The two regions are identified through a binary mask encoded by means of context-coding with adaptive template selection. Two coding modalities are defined, namely "Intra" and "Inter". In the proposed "Intra" mode no information about previously encoded macroblocks is exploited. When the proposed "Inter" mode is selected, previously encoded edge macroblocks in the same slice are exploited to improve the compression performance of context-coding and achieve significant bit rate savings while preserving edge information. Experimental results show that the proposed method implemented on the H.264/AVC framework can outperform a number of state-of-the-art edge-aware depth coding methods in terms of depth coding, view synthesis performance versus depth bit rate, and view synthesis performance versus texture plus depth bit rate, and presents promising performance compared with more complex solutions for intra depth coding. Average Bjøntegaard bit rate reductions over a standard H.264/AVC coder of about 31% and 21% are noticed in terms of synthesized view quality versus depth bit rate and synthesized view quality versus texture plus depth bit rate, respectively. The proposed "Inter" mode requires on average about half the amount of bits required by the proposed "Intra" mode, highlighting the benefits of inter-macroblock edge coding.

References

[1] ISO/IEC JTC1/SC29/WG11, Applications and Requirements on 3D Video Coding, Doc. N12035, Geneva, Switzerland, Mar. 2011.

[2] ISO/IEC JTC1/SC29/WG11, Press Release of the 103rd Meeting in Geneva, Switzerland, Doc. N13253, Geneva (CH), Jan. 2013.

[3] T. Wiegand, G. Sullivan, G. Bjøntegaard, A. Luthra, Overview of the H.264/AVC video coding standard, *EEE Trans. Circuits Syst. Video Technol.* 13 (2003) 560–576.

[4] G. Sullivan, J. Ohm, W.-J. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) standard, *EEE Trans. Circuits Syst. Video Technol.* 22 (2012) 1649–1668.

[5] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, R. Tanger, Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability, *Signal Process., Image Commun.* 22 (2007) 217–234.

[6] K. Müller, P. Merkle, T. Wiegand, 3-D video representation using depth maps, *Proc. IEEE* 99 (2011) 643–656.

[7] G. Cheung, W.-S. Kim, A. Ortega, J. Ishida, A. Kubota, Depth map coding using graph based transform and transform domain sparsification, in: *Proc. 2011 IEEE Int'l Work. Multimedia Signal Process. (MMSP 2011)*, pp. 1–6.

[8] B. T. Oh, H.-C. Wey, D.-S. Park, Plane segmentation based intra prediction for depth map coding, in: *Proc. 2012 Picture Coding Symp. (PCS 2012)*, pp. 41–44.

[9] M. Zamarin, M. Salmistraro, S. Forchhammer, A. Ortega, Edge-preserving intra depth coding based on context-coding and H.264/AVC, in: *Proc. 2013 IEEE Int'l Conf. Multimedia Expo (ICME 2013)* (accepted).

[10] Y. Morvan, D. Farin, P. de With, Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images, in: *Proc. 2007 IEEE Int'l Conf. Image Process. (ICIP 2007)*, pp. 105–108.

[11] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. de With, T. Wiegand, The effects of multiview depth video compression on multiview rendering, *Signal Process., Image Commun.* 24 (2009) 73–88.

[12] L. Lucas, N. Rodrigues, C. Pagliari, E. da Silva, S. de Faria, Efficient depth map coding using linear residue approximation and a flexible prediction framework, in: *Proc. 2012 IEEE Int'l Conf. Image Process. (ICIP 2012)*, pp. 1305–1308.

[13] D. Graziosi, N. Rodrigues, C. Pagliari, E. da Silva, S. de Faria, M. Perez, M. de Carvalho, Multiscale recurrent pattern matching approach for depth map coding, in: *Proc. 2010 Picture Coding Symp. (PCS 2010)*, pp. 294 –297.

[14] S. Shimizu, H. Kimata, S. Sugimoto, N. Matsuura, Block-adaptive palette-based prediction for depth map coding, in: *Proc. 2011 IEEE Int'l Conf. Image Process. (ICIP 2011)*, pp. 117–120.

[15] G. Shen, W.-S. Kim, A. Ortega, J. Lee, H. Wey, Edge-aware intra prediction for depth-map coding, in: *Proc. 2010 IEEE Int'l Conf. Image Process. (ICIP 2010)*, pp. 3393–3396.

[16] C. Lan, J. Xu, F. Wu, Improving depth compression in HEVC by pre/post processing, in: *Proc. 2012 Int'l Work. Hot Topics 3D Multimedia (Hot3D 2012)*, pp. 611–616.

[17] W.-S. Kim, S. Narang, A. Ortega, Graph based transforms for depth video coding, in: *Proc. 2012 IEEE Int'l Conf. Acoustics, Speech, Signal Process. (ICASSP 2012)*, pp. 813–816.

[18] G. Cheung, A. Kubota, A. Ortega, Sparse representation of depth maps for efficient transform coding, in: *Proc. 2010 Picture Coding Symp. (PCS 2010)*, pp. 298–301.

[19] G. Cheung, J. Ishida, A. Kubota, A. Ortega, Transform domain sparsification of depth maps using iterative quadratic programming, in: *Proc. 2011 IEEE Int'l Conf. Image Process. (ICIP 2011)*, pp. 129–132.

[20] G. Shen, W.-S. Kim, S. Narang, A. Ortega, J. Lee, H. Wey, Edge-adaptive transforms for efficient depth map coding, in: *Proc. 2010 Picture Coding Symp. (PCS 2010)*, pp. 566–569.

[21] P. Merkle, C. Bartnik, K. Muller, D. Marpe, T. Wiegand, 3D video: depth coding based on inter-component prediction of block partitions, in: *Proc. 2012 Picture Coding Symp. (PCS 2012)*, pp. 149–152.

[22] S. Milani, P. Zanuttigh, M. Zamarin, S. Forchhammer, Efficient depth map compression exploiting segmented color data, in: *Proc. 2011 IEEE Int'l Conf. Multimedia Expo (ICME 2011)*, pp. 1–6.

[23] J. Ruiz-Hidalgo, J. R. Morros, P. Aflaki, F. Calderero, F. Marqués, Multiview depth coding based on combined color/depth segmentation, *J. Visual Commun. Image Repres.* 23 (2012) 42–52.

[24] I. Daribo, G. Cheung, D. Florencio, Arithmetic edge coding for arbitrarily shaped sub-block motion

prediction in depth video compression, in: *Proc. 2012 IEEE Int'l Conf. Image Process. (ICIP 2012)*, pp. 1541–1544.

[25] L. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, *ACM Trans. Graph.* 23 (2004) 600–608.

[26] M. Tanimoto, T. Fujii, K. Suzuki, View synthesis algorithm in view synthesis reference software 3.5 (VSRS 3.5), ISO/IEC JTC 1/SC 29/WG 11 Doc. M16090, May 2009.

[27] G. Bjøntegaard, Calculation of average PSNR differences between RD-curves, ITU-T SG 16 Q.6, VCEG-M33, Austin, Texas, USA, Apr. 2001.

[28] Y. Liu, Q. Huang, S. Ma, D. Zhao, W. Gao, Joint video/depth rate allocation for 3D video coding based on view synthesis distortion model, *Signal Process., Image Commun.* 24 (2009) 666–681.

[29] I. Daribo, C. Tillier, B. Pesquet-Popescu, Motion vector sharing and bitrate allocation for 3D video-plus-depth coding, *EURASIP J. Appl. Signal Process.* 2009 (2008) 1–13.

TEXTURE SIDE INFORMATION GENERATION FOR DISTRIBUTED CODING OF VIDEO-PLUS-DEPTH

Matteo Salmistraro[◊] *Lars Lau Rakêt*[⋆] *Marco Zamarin*[◊] *Anna Ukhanova*[◊] *Søren Forchhammer*[◊]

[◊]DTU Fotonik, Technical University of Denmark, Ørsteds Plads,
2800 Kgs. Lyngby, Denmark. Emails: {matsl, mzam, annuk, sofo}@fotonik.dtu.dk
[⋆]Department of Computer Science, University of Copenhagen, Universitetsparken 5,
2100 Copenhagen, Denmark. Email: larslau@diku.dk

ABSTRACT

We consider distributed video coding in a monoview video-plus-depth scenario, aiming at coding textures jointly with their corresponding depth stream. Distributed Video Coding (DVC) is a video coding paradigm in which the complexity is shifted from the encoder to the decoder. The Side Information (SI) generation is an important element of the decoder, since the SI is the estimation of the to-be-decoded frame. Depth maps enable the calculation of the distance of an object from the camera. The motion between depth frames and their corresponding texture frames (luminance and chrominance components) is strongly correlated, so the additional depth information may be used to generate more accurate SI for the texture stream, increasing the efficiency of the system. In this paper we propose various methods for accurate texture SI generation, comparing them with other state-of-the-art solutions. The proposed system achieves gains on the reference decoder up to 1.49 dB.

Index Terms— Distributed Video Coding, Depth Map, Wyner-Ziv Coding, Optical Flow, Multi-Hypothesis

1. INTRODUCTION

In the recent years Distributed Video Coding (DVC) has received a great amount of interest, due to the possibility of shifting complexity from the encoder to the decoder.

In this paper we address DVC of video-plus-depth streams in a monoview scenario and propose methods to exploit the correlation between the streams in order to produce more accurate Side Information (SI). Depth maps can be used in single view scenarios for activity detection, object tracking and background/foreground separation [1].

DVC is based on two information theoretic results, the Slepian-Wolf [2] and Wyner-Ziv [3] (WZ) theorems, where, in the second case, source data are independently lossy coded but jointly decoded using a correlated source at the decoder, commonly referred to as SI. DVC could be an appealing solution for the video-plus-depth coding problem, in particular if we require low-complexity encoders. It is possible, in this way, to independently code the two streams and then jointly decode them. This is especially convenient when separated texture and depth cameras are used, in which case inter-camera communication is difficult or perhaps infeasible. The DVC decoder used as basis of our system is the one presented in [4], employing the approach first proposed in [5] and then improved in [6]. As can be seen in Fig. 1, the frames are divided into Key-Frames (KFs) and WZ frames at the encoder. The KFs are encoded independently with respect to each other and with respect to the WZ frames, using a H.264/AVC Intra coder. The KFs are used at the decoder to calculate the SI, which is a prediction of the to-be-decoded WZ frame. At the encoder the WZ frame is DCT-transformed, the coefficients are grouped and divided in bitplanes. Each bitplane is encoded using an LDPCA encoder [7], and a subset of the calculated syndromes is sent to the decoder. The decoder uses the syndromes to correct the errors in the corresponding SI bitplanes, bitplane by bitplane. If the syndromes are not enough, others are requested via a feedback channel. The LDCPA decoder also requires the calculation of the reliability of the bits of the bitplanes. Ideally, it is possible to calculate such reliability from the residual, which is the difference between the SI and the original WZ frame, but since WZ frames are not available at the decoder, a residual estimation method have to be devised.

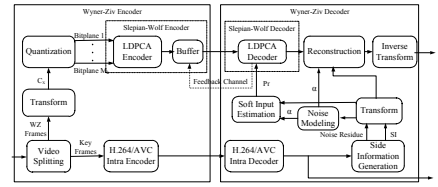


Fig. 1: DVC Codec [4].

Depth maps are images allowing the calculation of the distance of an object from the camera. While texture frames contain the luminance and chrominance components of the

M. Salmistraro, L.L. Rakêt, M. Zamarin, A. Ukhanova, S. Forchhammer, "Texture Side Information Generation for Distributed Coding of Video-Plus-Depth", *2013 IEEE Int'l Conf. on Image Proc. (ICIP 2013)*, Melbourne, Australia, Sep. 15-18, 2013 *(accepted)*.

(a) Texture Frame (b) Depth Frame

Fig. 2: A texture frame (a) and its correspoding depth frame (b), from the Ballet [12] sequence.

scene (Fig. 2a), depth maps describe depth information (Fig. 2b). Depth information can be used to calculate the distance of a given point in the 3D scene from the camera. The depth and texture frames of the same scene, referring to the same time instant, are strongly correlated and the motion between texture frames is highly correlated with the motion between the depth frames [8]. This gives rise to the video-plus-depth coding problem, in which the redundancy between the two streams is used to achieve efficient coding, as for example in [8]. This approach has also been used in depth map coding architectures based on DVC [9, 10] where the depth motion estimation has been carried out exploiting texture data. In DVC for texture frames, depth data have been used to generate intra-view SI through view synthesis [11]. View synthesis can be used in Multiview DVC but it is not suitable for single view systems or in the cases in which the Rate-Distortion (RD) performance of the intra and inter-view SIs are too different to obtain improvements from the fusion of the SIs.

This paper proposes methods for exploiting depth maps in the texture SI generation. We consider that an independently coded depth stream is already available and it is used to improve the WZ coding performance of the texture stream. We introduce Optical Flow (OF) based techniques, extending the framework proposed in [13] and introducing a new OF technique based on two distinct data terms. We benchmark these techniques against well-known block-based systems. Finally we consider the use of a multi-hypothesis decoder [14] for efficient and robust SI fusion. OF-based SI generation [15, 16] has been previously used in DVC as a way to create accurate SI for texture streams. In this paper we use OF to extract accurate motion estimates from the depth stream. We also propose a joint stream calculation, taking into account, at the same time, both KFs and depth frames, employing an OF formulation with two constraints.

2. SIDE INFORMATION GENERATION

Let D_i and T_i, with temporal index i, denote depth and texture frames, respectively. The to-be-decoded frame is T_t, all the other frames in Fig. 3 are assumed to be known at the decoder. We estimate the motion between D_t and D_{t-1} and use it to motion compensate T_{t-1} obtaining Y_{t-1}. We also calculate the motion between D_t and D_{t+1}, then T_{t+1} is motion compensated obtaining Y_{t+1}.

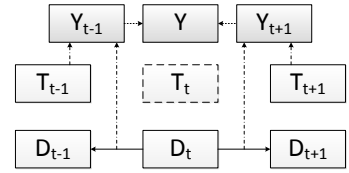


Fig. 3: The video stream structure, Group-Of-Pictures 2.

Once these two components have been calculated, the final SI Y can be calculated as their average, and the residual R can be calculated as their difference. We propose three new methods for this basic setup (Fig. 3).

The first two, "D2T BB" and "D2T OF", calculate the motion using the depth frames only, then this motion is used to motion compensate the texture frames, generating Y_{t-1} and Y_{t+1}. The difference between the two is the Motion Estimation (ME) algorithm: D2T BB uses a Block-Based (BB) method, while D2T OF uses an OF method.

For what concerns D2T BB, we consider the so-called "Adaptive Rood Pattern Search" (ARPS) ME algorithm [17]. While this approach may not provide the lowest Mean-Squared-Error between the motion compensated depth frame and the original one on average, it is able to capture the motion between the frames in a robust way, leading to fewer artefacts in the warped (texture) frame. ARPS has been proposed as a way to reduce the complexity of the ME process in state-of-the-art predictive coding, but thanks to the adaptive nature of the pattern and the refinement step, it produces superior results compared with full search ME in the given setup. The final method that we propose, "DT2T", is an OF method that uses both texture and depth information. This method employs the symmetric (texture) data term proposed in [13], but also adds the asymmetric information given by the depth maps in the motion estimation. In addition, we consider the symmetric texture based SI generation method presented in [16], which produces state-of-the-art results, as an alternative that does not use depth maps. This method is denoted as "T2T".

2.1. Optical Flow based SI generation

As opposed to BB motion estimation, OF gives a dense result, calculated by means of a global regularization process. Typical SI generation methods are based on calculating motion using texture KFs [14, 16]. Here we extend the symmetric OF method of [13] to also include asymmetric depth information. A novelty of our approach is the introduction of a new OF-based SI generation system, in which two data terms are jointly minimized.

Given a set of pixel-domain (texture) key frames and depth frames T_{t-1}, T_{t+1}, D_t, and $D_{t'}$, $t' = t-1$ or $t' = t+1$,

we want to estimate the dense flow field v such that the following optical flow constraints

$$C_T(\boldsymbol{x}, v) \triangleq T_{t+1}(\boldsymbol{x} + v(\boldsymbol{x})) - T_{t-1}(\boldsymbol{x} - v(\boldsymbol{x})), \quad (1)$$

$$C_D(\boldsymbol{x}, v) \triangleq D_{t'}(\boldsymbol{x} + v(\boldsymbol{x})) - D_t(\boldsymbol{x}), \quad (2)$$

are minimized, where $\boldsymbol{x}$ denotes a 2D point in the image.

The OF constraints are not sufficient for the motion estimation, and in order to make the problem well-posed, one has to penalize irregular behaviour. Here we focus on the TV-L^1 energy [18], where data fidelity between two frames is measured by L^1-norms of the optical flow constraints, and the global regularization term penalizes the total variation E of the estimated motion:

$$E(v) = \int \lambda_1 \|C_T(\boldsymbol{x}, v)\| + \lambda_2 \|C_D(\boldsymbol{x}, v)\| + \|\mathscr{D}v(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}. \quad (3)$$

With two data terms, this energy cannot be minimized as proposed in [13], unless $\lambda_1 = 0$ (D2T) or $\lambda_2 = 0$ (T2T). However, an extension to a sum of two 1-norm data terms (including the cases $\lambda_1 = 0$ or $\lambda_2 = 0$) is presented in [19]. This solution is used to substitute the original data term solution in [13], giving an algorithm that minimizes (3). The first data term produces flows that are symmetric through the interpolated frame, while the other term allows non-symmetric motion vectors. This combination should produce motion vectors where smaller details are matched using depth information, while bigger details (including lighting changes and shadows, which are not visible from depth data) should be matched using the texture frames. With the given formulation (3) we consider three distinct cases:

T2T: $\lambda_1 = 40$, $\lambda_2 = 0$,
D2T: $\lambda_1 = 0$, $\lambda_2 = 30$,
DT2T: $\lambda_1 = 5$, $\lambda_2 = 40$.

It has to be noted that DT2T can only be calculated by using the new OF introduced here, while D2T and T2T could have been calculated by using the method presented in [13]. The final estimate of the motion v is recovered following the general implementation described by [13], with the following exceptions: 65 pyramid levels are used, and 90 warps with 10 inner iterations are performed on each level; the Gaussian smoothing of input images prior to downsampling has standard deviation 0.5 for T2T and DT2T, and 0.35 for D2T; after linear upsampling of the flows, they are filtered using a 3×3 median filter. OF naturally leads to the non pixel location problem, in which a target position in T_{t-1} and T_{t+1} does not have integer coordinates. In this case bicubic interpolation is used.

2.2. Side Information Fusion

As previously outlined, our system is based on [4], and therefore also uses the Overlapped Block Motion Compensation (OBMC) SI. We propose the use of a multi-hypothesis decoder as a way of fusing the produced SIs [14]. For each SI the probability distribution of the bits of the bitplanes is calculated using the SI and its estimated residual. The distributions are then combined together using fixed weighting coefficients. Six different coefficients are used, and the resulting distributions are fed into six LDPCA decoders. For each new received chunk of syndromes the decoding is tried; if one of the decoders converges, its result is taken as final result and its combined distribution is used to reconstruct the corresponding DCT coefficients. This process can be also seen as a decoder-based rate-optimization since the chosen solution is the one requiring less bits. We employed the 2 SIs decoder (denoted as "2SI") and a 3 SIs decoder (denoted as "3SI"). For the 2SI decoder the first SI is OBMC and the second is chosen between the ones presented here. For the 3SI decoder we use OBMC, DT2T and T2T as SIs.

Table 1: QPs used with the given quantization matrices Q_i.

Sequence \ $\mathbf{Q_i}$	Q_1	Q_4	Q_7	Q_8
Dancer	33	30	27	23
Ballet	38	31	24	19
Breakdancers	40	33	26	22

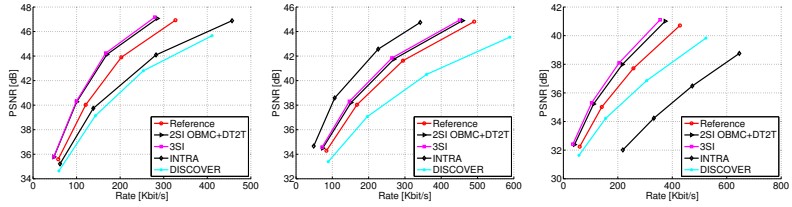
3. EXPERIMENTAL RESULTS

The system has been tested on a single view of the sequences "Breakdancers" and "Ballet" from Microsoft Research [12], and "Dancer" from Nokia Research [20]. We used the central view of the three sequences, at 15 fps downsampled to CIF resolution. The quantization matrices Q_i, $i = 1, 4, 7, 8$ of the DISCOVER project [21] are employed. The (texture) KFs are H.264/AVC Intra encoded using the QPs in Table 1. We have tested the first 100 frames of each sequence and reported the results for Group-Of-Pictures (GOP) 2, using as reference the decoder presented in [4] on texture frames. All the results and the graphs show only the WZ frames performance, since the KFs are encoded in the same manner for all the sequences. The rate of the depth frames is not taken into account, since we suppose that they are already required by the system and not only used to improve the coding performance.

The Bjøntegaard PSNR distances and bit-rate savings [22] for the 2SI decoder have been reported in Table 2, using uncompressed depth maps (denoted as $\mathrm{QP}_D = U$), and H.264/AVC Intra coded depth maps with quantization parameter $\mathrm{QP}_D = \{20, 40\}$. In Table 2, each 2SI decoder is denoted with the second employed SI, since the first one is always OBMC. DT2T is an extension of T2T, hence we report also the latter for the ease of comparison. From Table 2, in the case of uncompressed depth maps, we can see that DT2T is the best performing method for Dancer and Ballet, both medium motion sequences. For Ballet the second best method is D2T OF, while for Dancer the difference between D2T OF and T2T is negligible. It has to be noted that Ballet is a real-world sequence and depth maps have been estimated from texture data. Dancer, on the other hand, is a computer-

Table 2: Bjøntegaard Distances between the reference decoder [4] and the proposed decoders.

Sequence		T2T	$QP_D = U$			$QP_D = 20$			$QP_D = 40$		
			D2T BB	D2T OF	DT2T	D2T BB	D2T OF	DT2T	D2T BB	D2T OF	DT2T
Dancer	ΔRate[%]	15.44	11.12	17.57	23.82	9.48	17.16	24.55	5.41	12.57	19.29
	ΔPSNR[dB]	0.74	0.48	0.78	1.12	0.42	0.76	1.15	0.23	0.56	0.88
Ballet	ΔRate[%]	2.76	11.03	17.42	19.11	9.46	17.69	18.97	6.20	15.36	17.03
	ΔPSNR[dB]	0.19	0.74	1.19	1.32	0.63	1.22	1.32	0.41	1.05	1.17
Break-dancers	ΔRate[%]	3.84	8.50	12.43	11.61	7.71	12.30	11.83	5.03	11.15	10.95
	ΔPSNR[dB]	0.24	0.52	0.76	0.71	0.47	0.75	0.73	0.30	0.68	0.67



(a) Ballet (b) Breakdancers (c) Dancer

Fig. 4: RD curves, WZ frames only, uncompressed depth maps.

generated sequence in which depth maps have been generated using the actual distances of the 3D object models from the virtual camera. The depth maps of Dancer are smoother compared with those of Ballet, hence the SI of Dancer has lower quality compared with Ballet. Nevertheless, the novel DT2T approach outperforms both D2T OF and T2T, improving over the single SI decoder [4] by up to 1.32 dB. Breakdancers shows a much higher temporal activity making the motion estimation more difficult. In this case D2T OF greatly outperforms T2T. DT2T shows a negligible performance loss compared with D2T OF. In all the aforementioned cases D2T BB is not able to achieve the same performance as D2T OF due to the lack of flexibility of the block-based approach. The proposed OF-based methods show high resilience to the quantization noise of the depth maps: the performance in the case of $QP_D = 20$ are basically the same as in the uncompressed case; while in the case of $QP_D = 40$ we can notice a performance degradation, but the DT2T method is still able to achieve improvements ranging from 0.67 to 1.17 dB over [4]. It may also be noted that DT2T works correctly even when T2T has better performance compared with D2T OF (see Dancer, $QP_D = 40$) and it is still superior to the best of them. For what concerns the 3SI decoder (Table 3), it is able to correctly fuse the SIs leading to good and robust improvements, always superior to any 2SI decoder, with gains ranging from 0.90 dB to 1.49 dB. No case-specific optimization has been performed, i.e. the parameters used for the OF-based methods are fixed for all the sequences and for all the QP_D values. The RD-curves for the three sequences in the case of uncompressed texture frames are also depicted in Fig. 4,

Table 3: Bjøntegaard Distances between the reference decoder [4] and the 3SI decoder.

Sequence \ QP_D		U	20	40
Dancer	ΔRate[%]	30.69	30.80	28.40
	ΔPSNR[dB]	1.48	1.49	1.35
Ballet	ΔRate[%]	20.70	20.63	18.96
	ΔPSNR[dB]	1.45	1.45	1.32
Break-dancers	ΔRate[%]	15.15	15.05	14.49
	ΔPSNR[dB]	0.94	0.93	0.90

where the performance of [4] is denoted as "Reference". As it can be seen the decoder in [4] is able to greatly outperform the DISCOVER [6] decoder on textures in all the settings, making it more fair to compare the proposed systems with the one in [4].

4. CONCLUSION

In this work we investigated the possibility of using depth maps for improved SI generation in single-view video-plus-depth DVC. The proposed system is able to achieve good and robust improvements over one of the best single SI DVC decoders available in literature [4], with improvements ranging from 0.90 dB to 1.49 dB. OF-based methods showed clear superiority to conceptually similar block-based methods. The DT2T method was able to successfully combine the symmetrical OF approach [16] and the D2T approach introduced in this work. Finally, the multi-hypothesis decoder was able to successfully and robustly fuse the SIs here presented.

5. REFERENCES

[1] S. Mehrotra, Z. Zhang, Q. Cai, C. Zhang, and P.A. Chou, "Low-complexity, near-lossless coding of depth maps from Kinect-like depth cameras," in *Proc. of IEEE MMSP*, October 2011, pp. 1 –6.

[2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, pp. 471 – 480, July 1973.

[3] A.D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, pp. 1–10, 1976.

[4] X. Huang and S. Forchhammer, "Cross-band noise model refinement for transform domain Wyner-Ziv video coding," *Signal Processing: Image Communication*, vol. 27, no. 1, pp. 16 – 30, 2012.

[5] B. Girod, A.M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," in *Proc. of the IEEE*, January 2005, vol. 93, pp. 71 –83.

[6] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: Architecture, techniques and evaluation," in *Proc. of PCS*, November 2007.

[7] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing Journal*, vol. 86, no. 11, pp. 3123–3130, November 2006.

[8] M. Winken, H. Schwarz, and T. Wiegand, "Motion vector inheritance for high efficiency 3D video plus depth coding," in *Proc. of PCS*, May 2012, pp. 53 –56.

[9] G. Petrazzuoli, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu, "Wyner-Ziv coding for depth maps in multiview video-plus-depth," in *Proc. of IEEE ICIP*, September 2011, pp. 1817 –1820.

[10] M. Salmistraro, M. Zamarin, L. L. Rakêt, and S. Forchhammer, "Distributed multi-hypothesis coding of depth maps using texture motion information and optical flow," in *Proc. of IEEE ICASSP*, May 2013, *accepted*.

[11] X. Artigas, E. Angeli, and L. Torres, "Side information generation for multiview distributed video coding using a fusion approach," in *Proc. of NORSIG 2006*, June 2006, pp. 250 –253.

[12] L.C. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, 2004.

[13] L.L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert, "Motion compensated frame interpolation with a symmetric optical flow constraint," in *Advances in Visual Computing*, George Bebis *et al.*, Ed., vol. 7431 of *Lecture Notes in Computer Science*, pp. 447–457. Springer Berlin Heidelberg, 2012.

[14] X. Huang, L.L. Rakêt, H.V. Luong, M. Nielsen, F. Lauze, and S. Forchhammer, "Multi-hypothesis transform domain Wyner-Ziv video coding including optical flow," in *Proc. of IEEE MMSP*, October 2011, pp. 1 –6.

[15] H.V. Luong, L.L. Raket, X. Huang, and S. Forchhammer, "Side information and noise learning for distributed video coding using optical flow and clustering," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4782 –4796, December 2012.

[16] L.L. Rakêt, J. Søgaard, M. Salmistraro, H. V. Luong, and S. Forchhammer, "Exploiting the error-correcting capabilities of low density parity check codes in distributed video coding using optical flow," in *Proc. of SPIE*, 2012, vol. 8499, pp. 84990N–84990N–15.

[17] Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1442 – 1449, December 2002.

[18] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L^1 optical flow," in *Ann. Symp. German Association Patt. Recogn*, 2007, pp. 214–223.

[19] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers, "Duality TV-L^1 flow with fundamental matrix prior," in *Image and Vision Computing*, Auckland, New Zealand, November 2008, pp. 1–6.

[20] "Extension of existing 3DV test set toward synthetic 3D video content," ISO/IEC JTC1/SC29/WG11, Doc. M19221, Daegu, Korea, January 2011.

[21] "DISCOVER project test conditions," December 2007, http://www.img.lx.it.pt/~discover/test_conditions.html.

[22] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *VCEG Meeting*, Austin, USA, April 2001.

EDGE-PRESERVING INTRA DEPTH CODING BASED ON CONTEXT-CODING AND H.264/AVC

Marco Zamarin[1], *Matteo Salmistraro*[1], *Søren Forchhammer*[1], *Antonio Ortega*[2]

[1] Dept. of Photonics Eng., Technical University of Denmark, Denmark
[2] Ming Hsieh Dept. of Electrical Eng., University of Southern California, CA (USA)
{mzam, matsl, sofo}@fotonik.dtu.dk, ortega@sipi.usc.edu

ABSTRACT

Depth map coding plays a crucial role in 3D Video communication systems based on the "Multi-view Video plus Depth" representation as view synthesis performance is strongly affected by the accuracy of depth information, especially at edges in the depth map image. In this paper an efficient algorithm for edge-preserving intra depth compression based on H.264/AVC is presented. The proposed method introduces a new Intra mode specifically targeted to depth macroblocks with arbitrarily shaped edges, which are typically not efficiently represented by DCT. Edge macroblocks are partitioned into two regions each approximated by a flat surface. Edge information is encoded by means of context-coding with an adaptive template. As a novel element, the proposed method allows exploiting the edge structure of previously encoded edge macroblocks during the context-coding step to further increase compression performance. Experiments show that the proposed Intra mode can improve view synthesis performance: average Bjøntegaard bit rate savings of 25% have been reported over a standard H.264/AVC Intra coder.

Index Terms— Block-based depth compression, context-coding, edge-based depth representation, video-plus-depth, depth-image-based-rendering.

1. INTRODUCTION

In the recent years the interest in three-dimensional (3D) video technologies has grown considerably in both the academic and industrial worlds. A number of 3D-capable solutions and products are becoming available on the consumer market. One of the key challenges in the implementation of a 3D Video communication system is the decoupling of the capture and transmission format from the display format in order to allow a multitude of acquisition, transmission, and display devices to work together seamlessly [1]. One representation that enables such decoupling is the so-called "Multi-view Video plus Depth" (MVD), in which depth or disparity information is provided together with typically 2 or 3 views. By using depth information together with the input views, the desired output views can be synthesized at the decoder side thus allowing different 3D display devices to operate properly. Efficient coding solutions based on the MVD format are currently being developed by the 3DV group of MPEG [2]. Solutions compatible with both the current H.264/AVC and the upcoming High Efficiency Video Coding (HEVC) standards are being investigated.

As view synthesis algorithms based on Depth-Image-Bases-Rendering (DIBR) typically show a high sensitivity to depth inaccuracies [3], depth coding plays a crucial role in the development of an effective 3D Video system based on the MVD format. Specifically, depth edges should be preserved in order to avoid the appearance of annoying unnatural artifacts in the synthesized views. Due to the fact that standard DCT-based approaches fail to efficiently represent sharp edges, a number of specialized algorithms have been proposed in the literature to cope with this problem, e.g. based on edge-adaptive transforms and transform domain sparsification [4] or local explicit coding of edge information in blocks with sharp discontinuities [5, 6]. An overview of some recent works is provided in the next section.

In this paper a novel algorithm for efficient edge-aware intra depth coding is presented. The proposed scheme is based on the H.264/AVC Intra framework and operates at a MacroBlock (MB) level. While DCT fails to efficiently represent blocks with arbitrarily shaped edges, it provides very compact representations in the case of blocks with (nearly-)uniform values or smooth gradients. Therefore, we propose to modify the reference encoder by adding one Intra mode specifically targeted to edge MBs. When the new mode is selected, the MB is partitioned into two regions and a constant value is assigned to each region and encoded. Partitioning information also needs to be transmitted in order to allow for a proper reconstruction. This is done by encoding a per-pixel binary mask by means of context-coding. The proposed algorithm allows exploiting previously-encoded edge MBs in order to predict the constant values of the current MB being encoded. Moreover, the binary masks of adjacent edge MBs can be combined and jointly context-coded to allow further bit rate savings.

M. Zamarin, M. Salmistraro, S. Forchhammer, A. Ortega, "Edge-preserving Intra Depth Coding based on Context-coding and H.264/AVC", *2013 IEEE Int'l Conf. on Multimedia and Expo (ICME 2013)*, San Jose, CA, USA, July 15-19, 2013 *(accepted)*.

The remainder of this paper is organized as follows. In Section 2 related works on edge-based depth compression are briefly discussed. Section 3 describes the proposed coding method highlighting the novelties it introduces. Section 4 discusses the coding performance of the proposed framework on a number of 3D video test sequences.

2. RELATED WORK

As mentioned in Sec. 1, a number of edge-preserving depth coding algorithms have recently been proposed in the literature. Most of them are motivated by the fact that edge information is critical for the achievement of high view synthesis performance and therefore should be preserved in the coding process.

Shen *et al.* [7] introduced a set of edge-adaptive transforms for block-based depth coding. Edge detection was used to identify depth discontinuities and define a local graph structure to be transformed. Even though some view synthesis improvements were achieved, the explicit calculation of the eigenvectors of the graph Laplacian made the algorithm not suitable for fast implementations. Cheung *et al.* [8, 9] introduced the concept of *don't care region*, exploited to obtain sparse depth representations and improve the coding performance of a JPEG coder. The concept was then combined with the edge-adaptive transforms previously mentioned to achieve further improvements [4]. In [10] Kim *et al.* proposed to use graph based transforms as an alternative to DCT for 4×4 edge blocks within the H.264/AVC framework. Significant bit rate savings are reported, but again the calculation of the graph Laplacian caused a considerable increase of the overall algorithm complexity. Morvan *et al.* [11] proposed to extend the wedgelets - which locally approximate a signal with two constant functions separated by a straight line - with piecewise-linear functions (platelets) defined using a quadtree decomposition, reporting good results in terms of depth compression and edge preservation.

Block-based depth coding exploiting co-located texture edge information has been proposed by Merkle *et al.* in [12] for the case of HEVC. When texture is exploited to bi-partite a depth edge block, a constant depth value is assigned to each partition. As the same depth partitioning can be reproduced at the decoder side, only the two constant values need to be encoded. Depth image coding exploiting texture edge information has also been proposed by Milani *et al.* in [13]. In this case, segmentation of reconstructed texture data is exploited to predict shapes in the depth image. Within each segment the depth signal is approximated by a linear function and only the corresponding coefficients are transmitted, thus avoiding the explicit encoding of edge information.

Shimizu *et al.* [5] proposed a depth coding scheme based on H.264/AVC similar to the one proposed in this paper: edge macroblocks are approximated by a palette with two entries and a (binary) object shape map, both predictively encoded exploiting intra/inter correlation. Shape maps are generated by minimizing a prediction error function and encoded by means of context-adaptive binary arithmetic coding as in shape coding in MPEG-4 Part 2. Even though intra/inter neighboring blocks are exploited in order to properly define the context at any pixel position, the edge structure of previously encoded edge macroblocks is not directly exploited to improve the coding efficiency of context coding. If the edge structure correlation of neighboring edge macroblocks is taken into account as in the method proposed in this paper, lower bit rates can be achieved for the coding of shape information, which can be of benefit especially at low bit rates.

A different approach based on a similar framework has been proposed for the case of HEVC by Lan *et al.* in [6]. In this case a wider range of block sizes are exploited, namely from 32×32 down to 4×4. Blocks are partitioned in up to 8 regions each approximated by a flat surface. Arithmetic coding is then employed to encode the region map and surface values. Significant gains are reported for the cases of both estimated and acquired depth sequences. However, higher performance are expected if the region map overhead is reduced. This can be achieved by exploiting the correlation between the edge structures of the current and previously encoded region maps during the encoding process as proposed in this work.

3. PROPOSED METHOD

As mentioned in Sec. 1, a new Intra mode for H.264/AVC - referred to as "EDGE" in the following - able to efficiently represent arbitrarily-shaped edges is proposed. Differently from other approaches, the proposed mode aims at combining the encoding of depth discontinuities that span multiple adjacent MBs in order to increase the coding efficiency of edge information. The method operates at a MB level as part of the Rate-Distorion (RD) optimization strategy: the EDGE mode is tested together with the standard intra modes and the mode with the lowest RD cost is selected.

The EDGE mode partitions the whole depth MB into two regions and associates one Constant Value (CV) to each of them. A per-pixel binary map M is used to identify pixels belonging to the two regions. In order to allow a perfect reconstruction at the decoder side, the mask M is losslessly encoded by means of context-coding together with the two CVs. A detailed description of the EDGE algorithm for a depth MB is provided below (see Fig. 1).

1. The input depth MB is partitioned into two regions. A point in 3D is defined for each pixel from its spatial coordinates and depth value. 3D-points are connected with the closest (in an Euclidean sense) n neighbors, n being a parameter initially set to 1. If more than two disconnected components are formed, n is increased and the procedure iterated. A 16×16 binary mask M

is defined.

2. One CV is associated to each region of M by selecting the median value of the corresponding pixels in the input depth MB.[1]
3. The mask M is losslessly encoded by means of context-based arithmetic coding (see Subsection 3.1 for more details). Let R_M be the number of bits spent to encode M, R_{CV} the number of bits spent to encode the two CVs, and $R = R_M + R_{CV}$ the total number of bits.
4. The distortion D is computed as MSE between the input depth MB and the binary EDGE MB.
5. The EDGE RD cost is computed as follows: $RD^{intra}_{EDGE} = D + \lambda \cdot R$, λ being the same Lagrangian multiplier used in the RD cost calculation of the standard Intra modes.
6. If the left-neighboring MB is available and encoded as EDGE MB:
   (a) Steps 3-5 are repeated using the same CVs used in the left-neighboring MB. In this way no bits are spent to encode the CVs for the current MB. Let RD^{left1}_{EDGE} be the corresponding RD cost.
   (b) Steps 3-5 are repeated using the same CVs and coding parameters for the mask M used in the left-neighboring MB (see Subsection 3.1). Let RD^{left2}_{EDGE} be the corresponding RD cost.
7. As Step 6 but using the top-neighboring MB. If it is available and encoded as EDGE MB, the costs RD^{top1}_{EDGE} and RD^{top2}_{EDGE} are defined.
8. The RD_{EDGE} cost is defined as the lowest cost among RD^{left1}_{EDGE}, RD^{left2}_{EDGE}, RD^{top1}_{EDGE}, RD^{top2}_{EDGE} (if available), and RD^{intra}_{EDGE}.

If RD_{EDGE} is lower than the RD cost obtained with standard H.264/AVC Intra coding, the EDGE mode is selected and the corresponding data (mask M, CVs and coding parameters) are included in the bit stream. At the decoder side, if the usage of the EDGE mode is detected, the binary mask is decoded by means of context-based decoding, CVs are decoded from the bit stream and the output EDGE MB is produced. If the EDGE mode is not used, standard Intra decoding is performed.

3.1. Binary mask encoding

The encoding of a binary mask M is done by means of context-based arithmetic coding. The encoding can be done in two different ways: *intra* (i.e. without exploiting previously-encoded EDGE MBs) and *inter* (i.e. exploiting previously-encoded EDGE MBs in the same slice). The two methods are described here in detail.

[1]The choice of the median over the mean - even though not necessarily optimal from a MSE point of view - reduces the influence of outliers, i.e. noisy pixels in the MB partition.

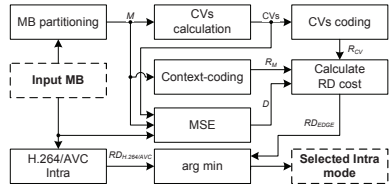


Fig. 1: Encoder block scheme

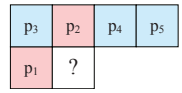


Fig. 2: Template selection: pixels p_1 and p_2 are fixed; one additional pixel among p_3, p_4, and p_5 can be included in the template. "?" indicates the current pixel being encoded.

3.1.1. *Intra EDGE coding*

In order to make the context prediction efficient, it is important to select the most relevant template pixels for the edge structure of the mask M being encoded. For this reason, an adaptive template inspired by [14] is used. The template is defined among a set of 5 candidate pixels (see Fig. 2) and has thus a smaller size than the one used in [14] and in MPEG-4 Part 2 shape coding. As shown in Fig. 2, the template includes the pixels on the left and on top of the one being encoded. A third pixel chosen among a set of three candidates can be included if of benefit in terms of code length: the binary mask is encoded 4 times - with the 2-pixel template and with the three 3-pixel templates - and the template providing the lower rate is selected.

In order to allow the decoder to correctly decode the mask M, the selected template is explicitly signalled in the bit stream. This is done by encoding two parameters: the number of pixels in the template and, in case of a 3-pixel template, the third pixel index. The bits needed to encode these parameters (referred to as $R_\#$ and R_p, respectively) are also considered in the template selection process.

For the *intra* case, the EDGE rate R introduced in Section 3, Step 3 is therefore given by

$$R^{intra} = R_\# + R_p + R_M + R_{CV} + R^{i/p}_{flag}, \qquad (1)$$

where $R^{i/p}_{flag}$ is the rate of a flag encoded only when at least one between the left and top neighboring MBs is available and encoded as EDGE MB, to signal whether *intra* on *inter* EDGE coding is used.

3.1.2. *Inter EDGE coding*

As specified in Section 3, Steps 6 and 7, if the left or top neighboring MB is available and encoded as EDGE MB, the

current MB can be inter encoded. This can be done in two ways referred to as *partial inter coding* and *full inter coding*, corresponding to Steps 6a-6b, and 7a-7b, respectively.

In the first case the encoding is done as in the intra case with the only difference that the CVs are not calculated and encoded but simply copied from the predicting MB, under the assumption that adjacent parts of the same edge refer to the same background and foreground objects. In this case the EDGE rate is given by

$$R^{p.inter} = R_{\#} + R_p + R_M + R^{i/p}_{flag} + R^{p/f}_{flag} + R^{l/t}_{flag}, \quad (2)$$

where $R^{p/f}_{flag}$ is the rate of a flag specifying whether *partial* or *full* inter coding is used, and $R^{l/t}_{flag}$ is the rate of a flag indicating which of the two neighboring MBs is used for prediction when both of them are available and encoded as EDGE MBs.

When *full inter coding* is used, together with the CVs also the template pixels are copied from the predicting MB. Moreover, since the context-coding will be based on the same template, it is possible to use the edge statistics of the predicting MB - which are stored for each EDGE MB after the encoding of the mask M - as a starting point for the context-coding, thus allowing to exploit the statistics of the edge in the predicting MB. When *full inter coding* is used, adjacent blocks are therefore encoded as part of a unique binary mask. Note that edge statistics do not need to be explicitly transmitted as they can be generated during the decoding process. Finally, in order to properly initialize the boundaries of the binary mask M for the context coding step, the encoded binary mask of the predicting MB is used (the same is done for the *partial inter coding* too). In case of *full inter coding* the EDGE rate is given by

$$R^{f.inter} = R_M + R^{i/p}_{flag} + R^{p/f}_{flag} + R^{l/t}_{flag}. \quad (3)$$

The availability of these two inter modes allows for efficient representation of depth MBs that separate between the same background and foreground of the previous MB (i.e. they share the same CVs) and present different edge direction/structure (*partial inter coding*) or similar structure (*full inter coding*).

4. EXPERIMENTAL RESULTS

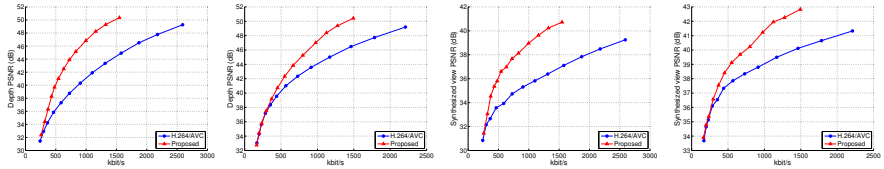
The proposed method has been implemented on the H.264/AVC reference software JM 17.1 and evaluated on the following 3D video sequences: *Ballet* and *Breakdancers* [15], *Book Arrival*, *Lovebird1*, *Dancer*, *Cafe*. The first 15 frames of each sequence have been encoded. Two views have been selected for each sequence (reported in Table 1) and the corresponding depth images have been encoded with both H.264/AVC Intra and the proposed method. Depth maps have been encoded with every second QP in the interval (24, 48). All the standard Intra modes of H.264/AVC have been enabled and RD-based mode decision was selected for the Intra 16×16 mode. In order to avoid introducing blurring artifacts, the deblocking filter has been disabled. In order to evaluate the effectiveness of the proposed algorithm, not only the RD performance on the depth signals have been evaluated (Figs. 3a and 3b for the sequences *Ballet* and *Breakdancers*), but also the view synthesis performance using reconstructed depth data and uncompressed texture as done in [16] (Figs. 3c and 3d) have been considered. In this case, PSNR values are measured against reference virtual views synthesized from uncompressed depth and texture data. Virtual views have been synthesized using the MPEG VSRS 3.5.

Table 1: Sequence resolutions and input left (L), virtual (V) and right (R) views.

Sequence	Resolution	L	V	R
Ballet	1024×768	5	4	3
Breakdancers	1024×768	5	4	3
Book Arrival	1024×768	8	7	6
Lovebird1	1024×768	6	7	8
Dancer	1920×1088	2	3	5
Cafe	1920×1088	2	3	4

Table 2 reports the Bjøntegaard bit rate savings [17] between the proposed method and H.264/AVC for both the depth signals and the sythesized views. As it can be noticed, when comparing the view synthesis performance the proposed method improves depth compression efficiency for all the sequences, confirming that it is of benefit in a MVD scenario. However, the improvement depends on the depth accuracy of the particular test sequence and on its resolution. Major bit rate savings in terms of both depth coding and view synthesis performance are reported in the case of *Ballet* and *Breakdancers* due to the sharp and clean edges that characterize depth images of these two sequences. Experimental results show that the proposed *inter EDGE coding* is often selected: the binary masks of adjacent edge macroblocks are therefore jointly encoded as part of a single binary image allowing additional bit rate savings (see Fig. 4b for an example). *Book Arrival* shows a comparable bit rate reduction in terms of view synthesis, but a smaller gain in terms of depth compression due to the presence of less sharp object discontinuities in the depth images which favor DCT coding. In the case of *Lovebird1*, a smaller gain in terms of synthesis performance is observed, together with a minor increase of the depth bit rate. This sequence shows that even though the proposed algorithm might not always be more efficient than H.264/AVC in terms of depth MSE, it does provide a benefit when reconstructed depth data are used for view synthesis.

The performance on the two high definition sequences are also satisfactory but lower gains are noticed, even for the *Dancer* sequence which features high quality synthetic depth data. One reason can be found in the fact that edge mac-

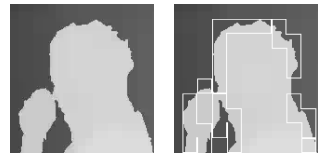


(a) *Ballet*: depth PSNR vs. depth bit rate (b) *Breakdancers*: depth PSNR vs. depth bit rate (c) *Ballet*: synth. view PSNR vs. depth bit rate (d) *Breakdancers*: synth. view PSNR vs. depth bit rate

Fig. 3: Rate-Distortion performance comparisons

Table 2: Bjøntegaard bit rate savings (BDBR) between the proposed method and H.264/AVC for depth signals and corresponding synthesized views. Note that the minus sign means bit rate reduction.

Sequence	Depth BDBR (%)	Synth. BDBR (%)
Ballet	-38.33	-46.82
Breakdancers	-21.94	-34.54
Book Arrival	-7.91	-34.39
Lovebird1	1.45	-10.78
Dancer	-3.60	-20.84
Cafe	0.45	-4.21
Average	**-11.65**	**-25.26**



(a) (b)

Fig. 4: Reconstructed depth comparison: detail of the Ballet sequence, left view, frame 0. (a) H.264/AVC Intra, 38.85 dB @ 361 kbit/s, (b) Proposed, 43.83 dB @ 366 kbit/s. White squares indicate MBs that have been encoded as EDGE MBs. EDGE MBs encoded by means of EDGE prediction are connected to the corresponding prediction MBs.

roblocks in high resolution depth images present on average less arbitrarily shaped discontinuities. Therefore, the standard H.264/AVC Intra modes can predict well the edge structures most of the times. For high resolution data, bigger block sizes - such as those defined in the upcoming HEVC standard - would hence be beneficial for the proposed algorithm especially at very low bit rates, where the overhead due to DCT/EDGE signalling can significantly affect the RD performance (as in the *Dancer*, *Cafe*, *Book Arrival*, and *Lovebird1* sequences in which H.264/AVC slightly outperforms the proposed method at very low bit rates).

Figures 4 and 5 provide a visual quality comparison between H.264/AVC Intra and the proposed method for both depth and synthesized data at similar bit rates, highlighting the advantages of the latter over the former.

Coding performance considering total bit rate (texture plus depth) versus synthesized view PSNR have also been evaluated, as done in [18]. Even though a direct comparison with the plane segmentation based coding method in [18] is not possible due to different reference settings and test conditions, the proposed method shows similar Bjøntegaard bit rate savings over the reference encoder for the common sequences, ranging between -11.50% and -3.98%.

Finally, Fig. 6 compares the RD performance of the proposed approach in terms of depth quality against H.264/AVC, JPEG2000, the method proposed by Milani *et al.* in [13], and the Platelet-based algorithm proposed by Morvan *et al.* in [11]. The comparison is made on the *Teddy* depth image from the stereo sets at [19]. As it can be noticed, the proposed method is able to outperform H.264/AVC Intra, the planar fitting method proposed in [13], which exploits reconstructed texture data in order to predict depth shapes, and the Platelet-based scheme. JPEG2000 appears as the least efficient still image coder in the case of depth images due to the heavy ringing artifacts it introduces.

Compared with Shimizu *et al.* [5] in which the number of bits needed to encode a shape/binary mask is around 50 per MB, the proposed method exhibits a more flexible behavior thanks to the *intra/inter EDGE coding* methods. The total number of bits needed to encode an EDGE MB (including binary mask, constant values, and flags) ranges from an average of 51 at high rates to an average of 45 at low rates, where *inter EDGE coding* is often selected.

5. CONCLUSIONS AND FUTURE WORK

In this paper a novel edge-preserving H.264/AVC Intra mode for efficient depth coding has been presented. Edge macroblocks are partitioned in two regions each approximated

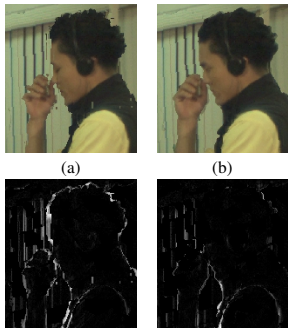


Fig. 5: Visual quality comparison versus total depth bit rate: detail of the synthesized view from uncompressed texture, Ballet sequence, frame 0. H.264/AVC Intra, 34.73 dB @ 726 kbit/s (a), Proposed, 37.67 dB @ 723 kbit/s (b), absolute errors on the luminance components (with same scale) between synthesized views obtained with uncompressed depth data and H.264/AVC Intra (c) and Proposed (d).

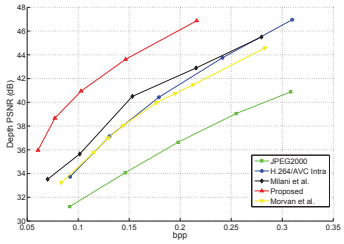


Fig. 6: *Teddy* depth image: RD performance comparison

by a flat surface. A binary mask identifying the two regions is defined and encoded by means of adaptive context-coding exploiting previously encoded edge macroblocks to increase the compression efficiency. Experimental results show that adjacent edge macroblocks can be jointly encoded in an efficient manner. Compared with a standard H.264/AVC Intra coder, the proposed algorithm achieved average Bjøntegaard bit rate savings of about 12% in terms of depth compression, and about 25% in terms of synthesized view quality versus depth bit rate. The complexities of the proposed and standard Intra modes are comparable, thus the overall complexity is not affected significantly. Future developments include binary mask pre-filtering for efficient low bit rate coding, prediction from non-EDGE macroblocks and quantization of flat surface values, and extension to depth video coding.

6. REFERENCES

[1] L. Onural, *3D Video Technologies: An Overview of Research Trends*, Society of Photo Optical, 2010.

[2] ISO/IEC JTC1/SC29/WG11, "Applications and Requirements on 3D Video Coding," Doc. N12035, Geneva (CH), Mar. 2011.

[3] K. Müller, P. Merkle, and T. Wiegand, "3-D Video Representation Using Depth Maps," *Proc. of the IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[4] G. Cheung, W.-S. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth Map Coding using Graph based Transform and Transform Domain Sparsification," in *IEEE MMSP 2011*, Oct. 2011, pp. 1–6.

[5] S. Shimizu, H. Kimata, S. Sugimoto, and N. Matsuura, "Block-adaptive Palette-based Prediction for Depth Map Coding," in *IEEE ICIP 2011*, Sep. 2011, pp. 117–120.

[6] C. Lan, J. Xu, and F. Wu, "Improving Depth Compression in HEVC by Pre/Post Processing," in *2012 Int'l Work. on Hot Topics in 3D Multim. (Hot3D 2012)*, July 2012, pp. 611–616.

[7] G. Shen, W.-S. Kim, S.K. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive Transforms for Efficient Depth Map Coding," in *PCS 2010*, Dec. 2010, pp. 566–569.

[8] G. Cheung, A. Kubota, and A. Ortega, "Sparse Representation of Depth Maps for Efficient Transform Coding," in *Picture Coding Symposium (PCS) 2010*, Dec. 2010, pp. 298–301.

[9] G. Cheung, J. Ishida, A. Kubota, and A. Ortega, "Transform Domain Sparsification of Depth Maps using Iterative Quadratic Programming," in *IEEE ICIP 2011*, Sep. 2011, pp. 129–132.

[10] W.-S. Kim, S.K. Narang, and A. Ortega, "Graph based Transforms for Depth Video Coding," in *IEEE ICASSP 2012*, Mar. 2012, pp. 813–816.

[11] Y. Morvan, D. Farin, and P. de With, "Depth-Image Compression based on an R-D Optimized Quadtree Decomposition for the Transmission of Multiview Images," in *Proc. of IEEE ICIP 2007*, 2007.

[12] P. Merkle, C. Bartnik, K. Muller, D. Marpe, and T. Wiegand, "3D Video: Depth Coding based on Inter-component Prediction of Block Partitions," in *Picture Coding Symposium (PCS) 2012*, May 2012, pp. 149–152.

[13] S. Milani, P. Zanuttigh, M. Zamarin, and S. Forchhammer, "Efficient Depth Map Compression Exploiting Segmented Color Data," in *IEEE ICME 2011*, July 2011, pp. 1–6.

[14] M. Zamarin and S. Forchhammer, "Lossless Compression of Stereo Disparity Maps for 3D," in *2012 Int'l Work. on Hot Topics in 3D Multim. (Hot3D 2012)*, July 2012, pp. 617–622.

[15] L.C. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-Quality Video View Interpolation using a Layered Representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, 2004.

[16] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P.H.N. de With, and T. Wiegand, "The Effects of Multiview Depth Video Compression on Multiview Rendering," *Image Commun.*, vol. 24, no. 1-2, pp. 73–88, 2009.

[17] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD-Curves," VCEG-M33, Apr. 2001.

[18] B.T. Oh, H.-C. Wey, and D.-S. Park, "Plane segmentation based intra prediction for depth map coding," in *PCS 2012*, May 2012, pp. 41–44.

[19] "Repository vision.middlebury.edu: Stereo datasets," `http://vision.middlebury.edu/stereo`.

DISTRIBUTED MULTI-HYPOTHESIS CODING OF DEPTH MAPS USING TEXTURE MOTION INFORMATION AND OPTICAL FLOW

Matteo Salmistraro[◊] *Marco Zamarin*[◊] *Lars Lau Rakêt*[⋆] *Søren Forchhammer*[◊]

[◊]DTU Fotonik, Technical University of Denmark, Ørsteds Plads,
2800 Kgs. Lyngby, Denmark. Emails: {matsl, mzam, sofo}@fotonik.dtu.dk
[⋆]Department of Computer Science, University of Copenhagen, Universitetsparken 1,
2100 Copenhagen, Denmark. Email: larslau@diku.dk

ABSTRACT

Distributed Video Coding (DVC) is a video coding paradigm allowing a shift of complexity from the encoder to the decoder. Depth maps are images enabling the calculation of the distance of an object from the camera, which can be used in multiview coding in order to generate virtual views, but also in single view coding for motion detection or image segmentation. In this work, we address the problem of depth map video DVC encoding in a single-view scenario. We exploit the motion of the corresponding texture video which is highly correlated with the depth maps. In order to extract the motion information, a block-based and an optical flow-based methods are employed. Finally we fuse the proposed Side Informations using a multi-hypothesis DVC decoder, which allows us to exploit the strengths of all the proposed methods at the same time.

***Index Terms*—** Distributed Source Coding, Depth Map Coding, Wyner-Ziv Coding, Optical Flow, Distributed Video Coding.

1. INTRODUCTION

In this work we address the coding of depth maps, using DVC [1, 2] as basis of our coding architecture.

Depth maps are particular images enabling the calculation of the distance of an object from the camera. A video representation format that is gaining popularity is the so-called "video-plus-depth", where in addition to texture data (the luminance and chrominance information of the scene), per-pixel depth information is also provided [3, 4]. Depth data allows fast generation of virtual views using the so-called Depth-Image-Based-Rendering (DIBR) algorithms [4], which makes the video-plus-depth format suitable for 3DTV and free viewpoint system implementations [5]. Moreover, it can be used for a number of purposes that can be of interest in modern video surveillance scenarios such as scene matting, activity detection and object tracking [3].

DVC is a video coding paradigm that allows shifting the complexity from the encoder side to the decoder side due to the fact that Motion Estimation (ME)—which heavily contributes to the computational complexity in state-of-the-art video codecs—can be performed at the decoder. Typical DVC scenarios feature strict power consumption constraints at the transmitter side, requiring low-complexity encoders, while the requirements are less stringent at the decoder. A multi-camera video surveillance scenario is a good example of a system with such requirements [2]. In a typical DVC architecture [1] inter-coded frames (i.e. frames coded by means of motion estimation and compensation) are substituted by the so-called Wyner-Ziv (WZ) frames. WZ frames are encoded in a different manner: parity check data are calculated and transmitted. An Intra-coded frame is referred to as Key Frame (KF) and is encoded and transmitted as in traditional video coding. At the decoder side KFs are used to estimate WZ frames by means of ME. The estimated frame, called Side Information (SI), can be corrected using parity bits from the encoder. The SI generation algorithm is therefore of crucial importance as the quality of the estimated frames directly affects the amount of additional parity bits required, and consequently the Rate-Distortion (RD) performance of the system. The core part of the proposed decoder is the Transform Domain WZ (TDWZ) codec [6]. At the encoder the WZ frame is DCT transformed and quantized. Each DCT coefficient is organized in bitplanes, and for each bitplane a LDPCA [7] encoder calculates the parity bits. At the decoder the SI is generated using an interpolation-based technique, for example Overlapped Block Motion Compensation (OBMC) [6]. A subset of the parity bits are sent to the decoder. The decoder tries to correct the errors present in the corresponding bitplane of the SI using the parity bits. If the decoding is not successful new bits are requested. Another key element of the decoder is the noise modelling, which is important in order to provide the LDPCA decoder with the likelihood of the value of each bit. The errors present in the SI are modelled as Laplacian distributed errors. In order to calculate the distribution, an estimation of the residual is needed. The residual is the difference between the SI and the original frame, which can not be directly calculated in practice. For more information on

M. Salmistraro, M. Zamarin, L.L. Rakêt, S. Forchhammer, "Distributed Multi-Hypothesis Coding of Depth Maps using Texture Motion Information and Optical Flow", *Proc. of 2013 IEEE Int'l Conf. on Acoustics, Speech, and Signal Proc. (ICASSP 2013)*, pp. 1685-1689, Vancouver, Canada, May 26-31, 2013.

DVC coding the reader is referred to [1, 2, 6].

Since texture and depth represent different aspects of the same 3D scene, the two components show a high correlation [8]. In a video-plus-depth DVC scenario such correlation can be exploited to improve the overall coding efficiency e.g. by refining the depth SI generation using texture motion information.

In this paper Transform Domain WZ coding of depth maps is addressed in a mono-view video-plus-depth scenario. This scenario is interesting when addressed with DVC because two dependent streams can be independently encoded but dependently decoded. This approach can be generalized to a multi-camera scenario, where a depth camera and a texture camera are used together, making inter-camera cooperation difficult or not feasible. Texture data are supposed to be available at the decoder and are used to improve the WZ decoding of depth data. Three different SIs are generated and fused using a multi-hypothesis approach [9]. The first SI is generated by applying block-based texture motion vectors to the depth component; the second one is obtained by applying the texture optical flow to the depth component; finally the third one is generated by means of motion estimation from depth data only. The three SIs present different characteristics and provide accurate estimation of the to-be-decoded depth frame in different regions.

1.1. Related Works

The use of texture motion information for depth compression purposes has been explored in conventional predictive coding in [10] and more recently in [11]. The same concepts can be exploited in a DVC decoder for accurate SI generation, as done in [12] in which multiple decoded texture frames are used. In our work we suppose that the decoder has access to the corresponding texture frame of the to-be-decoded depth frame, while in [12] only the texture frames corresponding to the depth KFs are used. Moreover, we investigate optical-flow-based methods, while [12] investigate only block-based methods. The multi-hypothesis decoder employed is the same as in [9] where OBMC was used with block-based extrapolation and optical flow-based interpolation in order to improve a texture-based DVC decoder. We use the same approach in order to effectively fuse three different SIs.

A preliminary study of the aforementioned problem has been performed in [13] but only the block-based method was presented and no fusion technique was proposed.

Optical flow-based SI generation has already been used for example in [9]. In this case the flows were used to interpolate an unknown texture frame given the previous and successive texture frames. In our framework we use the flow to extract the motion information from texture frames.

The remainder of this paper is organized as follows: Section 2 describes the proposed SI generation algorithms and the relative SI fusion method. In Section 3 experimental results are discussed. Finally, Section 4 summarizes the presented work.

2. SI GENERATION AND FUSION

In this section we describe the two proposed SI generation algorithms for depth maps, exploiting texture motion information. We also analyse the employed fusion procedure. In addition a third SI based on OBMC [6] on depth video is included in the fusion procedure. This decoder is used as basis for evaluating the performance of the two texture-based SIs and the performance of the fusion procedure. It has to be noted however, that OBMC has not been devised for depth maps and it has not been modified in this work.

2.1. Texture-based SI generation algorithms

The main idea behind the proposed methods is that the motion of the texture is highly correlated with the one encountered in the depth data. For the to-be-decoded depth frame X at instant t, assume that the depth maps at instants $t-1$, and $t+1$ (D_{t-1} and D_{t+1}, respectively) are known. We can use the motion information of the texture to warp D_{t-1} and D_{t+1} towards X obtaining the SI Y. In order to perform the aforementioned procedure the texture frames at instants $t-1$, t, and $t+1$ (C_{t-1}, C_t, C_{t+1}, respectively) are available at the decoder. The Motion Vectors (MVs) are calculated from C_t to C_{t-1} and from C_t to C_{t+1}. The MVs are used in turn to motion compensate D_{t-1} and D_{t+1}, obtaining two depth SIs Y_1 and Y_2, respectively. The final SI, Y, is calculated as the arithmetic average of Y_1 and Y_2. The residual, R_Y, is calculated as the absolute difference between Y_1 and Y_2. The argument behind this simple choice is that if a region in X presents simple motion, it will be well predicted. Hence Y_1 and Y_2 will agree in the particular area, leading to low residual estimation. If on the contrary the two estimated frames disagree, the residual will be higher.

The methods used to calculate the motion from the texture data is of central importance to the SI quality. We have selected two different ME approaches.

2.2. Block-Based Side Information Generation

We consider the so-called "Adaptive Rood Pattern Search" (ARPS) ME algorithm proposed in [14]. This approach may not provide the lowest MSE (Mean-Squared-Error) between the motion compensated texture frame and the original one, however, it is able to capture the motion between the frames in a robust way, leading to fewer artefacts in the warped (depth) frame. ARPS has been proposed as a way to reduce the complexity of the ME process in state-of-the-art predictive coding, but thanks to the adaptive nature of the pattern and the refinement step, it produces superior results compared with

full search in the given setup. This Block-Based SI generation is referred as BB.

2.3. Optical Flow Side Information Generation

As an alternative to BB, we consider an Optical Flow (OF) [15] SI generation. As opposed to BB, the OF based ME is global, in the sense that individual motion vectors are estimated for every pixel. Given a set of texture frames C_t and $C_{t'}$, $(t' = t+1, t-1)$, in pixel domain, we want to estimate the dense flow field v such that the optical flow constraint

$$D(\boldsymbol{x}, v) \triangleq C_{t'}(\boldsymbol{x} + v(\boldsymbol{x})) - C_t(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x}$ denotes a point in the image, is close to zero.

The optical flow constraint (1) will not be sufficient for motion estimation, and in order to make the problem well posed, one has to penalize irregular behavior. Here we focus on the TV-L^1 energy, where data fidelity between two frames is measured by the L^1-norm of the optical flow constraint, and the regularization term penalizes the total variation of the estimated motion:

$$E(v) = \int \lambda \|D(\boldsymbol{x}, v)\| + \|\mathscr{D}v(\boldsymbol{x})\| \, \mathrm{d}\boldsymbol{x}. \tag{2}$$

The total variation of a vector valued function is not uniquely defined, and several definitions have been used for this problem [16, 17, 18]. Here we use the definition of [19], since this method does not suffer from the channel smearing (i.e. independent optimization of the two channels of the motion vectors, the x- and y-components) of other definitions.

The final estimate of the motion v is recovered from iteratively minimizing a linearized version of (2) using the duality based splitting of [16]. The minimization is performed in a coarse to fine pyramid. We use 65 pyramid levels wit a scaling factor of 1.05, and Gaussian blurring of C_t and $C_{t'}$ with standard deviation 0.5, and on each level we perform 90 warps, with 1 outer and 10 inner iterations [16]. Furthermore we remove outliers by performing a median filtering of the flow for each warp. The parameter λ was set to 480. Compared to optical flow based interpolation [9] this value may seem high, however for the given test setup with higher temporal and spatial resolution, as well as the direct knowledge of the texture state, this higher weight on data fidelity is adequate. For more details on the implementation we refer to [18]. OF may lead to the non pixel location problem, in which a target position in D_{t-1} and D_{t+1} does not have integer coordinates. In this case bicubic interpolation is used.

2.4. Side Information Fusion

In order to exploit all the presented SIs (BB, OF, OBMC) a robust fusion technique is needed. In [9] it has been demonstrated that a multi-hypothesis decoder can be used to effectively combine block-based and pixel-based motion estimation techniques. In our work, we use the three SI decoders approach (referred as 3SI) as a way to fuse the three SIs. The multi-hypothesis decoder allows implementing a rate-based optimization strategy by using a number of parallel LDPCA decoders. Each LDPCA decoder is fed with a different weighted combination of the conditional probabilities for a given bitplane, and the syndromes coming from the encoder. Each bitplane contains the co-located bits of a given DCT coefficient. The decoded sequence of the first converging decoder is chosen as solution, and the corresponding weights used to combine the SIs are also used in the reconstruction process to improve the PSNR of the decoded frame. This method, thanks to the multi-decoder structure, shows robust gain, good performance and is therefore employed in this work as the fusion technique. However, the 3SI approach will increase the complexity of LDPC decoding up to 6 times.

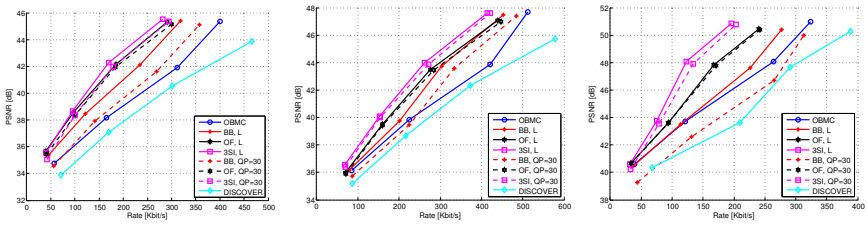
3. EXPERIMENTAL RESULTS

The system has been tested on the sequences "Breakdancers" and "Ballet" from Microsoft Research [20], and "Dancer" from Nokia Research [21]. We used the central view of the three sequences, at 15 fps downsampled to CIF resolution. The quantization matrices Qi = 1, 4, 7, 8 of the DISCOVER [22] project are employed. The KFs are H.264/AVC Intra encoded using QP = 40, 37, 31, 29 and are matched with the quantization matrices. We have tested the first 100 frames of each sequence and reported the results for Group-Of-Pictures (GOP) 2, 4, 8. The performance of the WZ frames has been evaluated and compared with the single SI OBMC decoder. In Tables 1-3 we list the Bjøntegaard differences [23] between the single SI OBMC decoder and the 3SI decoder. The results for lossless coded textures are listed as "QP = L", while the results using compressed textures, are listed with the QP used for compression. Texture compression has been performed with a standard H.264/AVC Intra coder[1]. In Figs. 1a-1c the RD curves for GOP2 are reported. We have also reported the performance of the single SI system and the performance of DISCOVER. Only the performance for WZ frames is reported. It has to be noted that the parameters of the 3SI decoder are the same for all the sequences and the quality level of the textures.

In Section 2, the SI generation for GOP2 has been outlined. In the cases of GOP4 and GOP8 a hierarchical coding structure [6] is used. First the SI for the central WZ frame is generated using C_{t-k}, C_{t+k}, D_{t-k}, and D_{t+k}, where k corresponds to half of the GOP size. The decoded WZ frame splits the GOP in two smaller GOPs in which the procedure can be iterated until all the WZ frames have been decoded.

From the results presented, it can be seen that the OF outperforms all the other single SI methods, showing also high robustness against texture quantization, while the BB

[1]JM 18.1 Reference Software, available at `iphome.hhi.de/suehring/tml`



(a) Ballet (b) Breakdancers (c) Dancer

Fig. 1: RD curves, WZ frames only, GOP2.

method suffers at lower qualities of the texture frames. The single SI OBMC-based decoder outperforms DISCOVER [24] codec in all the studied conditions and for all the investigated sequences. The 3SI is able to correctly fuse the three SIs, performing, on average, better or as well as the best available SI for the particular RD point. The improvements between the single SI OBMC decoder and the 3SI decoder ranges from 1.50 to 4.95 dB and from 21.24% to 49.06% bit-rate Bjøntegaard savings. Interestingly, the improvements for GOP8, are higher in the case of compressed textures for the Ballet and Breakdancers sequences (Table 3). A justification can be found in the non-linear low-pass filtering nature of the quantization, leading to more robust results, which in case of complex motion can be of benefit.

Sequence	**QP**	**ΔPSNR** [dB]	**ΔRate** [%]
Ballet	L	2.98	−46.46
	20	2.85	−44.99
	30	2.40	−39.79
Breakdancers	L	2.12	−34.02
	20	2.07	−33.28
	30	1.87	−31.16
Dancer	L	2.05	−42.90
	20	2.04	−40.41
	30	1.82	−36.24

Table 1: Bjøntegaard Distances between OBMC and the proposed methods, GOP2.

Sequence	**QP**	**ΔPSNR** [dB]	**ΔRate** [%]
Ballet	L	3.16	−44.71
	20	3.06	−43.32
	30	2.57	−38.11
Breakdancers	L	1.71	−23.87
	20	1.68	−23.52
	30	1.50	−21.24
Dancer	L	2.47	−42.54
	20	2.38	−42.03
	30	2.00	−38.81

Table 2: Bjøntegaard Distances between OBMC and the proposed methods, GOP4.

4. CONCLUSION

In this work we addressed the problem of DVC-based depth-map coding. We devised algorithms to produce higher quality SIs, employing the texture frames. We used two methods in order to extract the motion information from the texture frames: a block-based method and an optical flow-based one. The optical flow achieved better performance and superior robustness to quantization of the textures compared with the block-based system. The multi-hypothesis decoder proved to be an effective and robust way to fuse the three generated SIs outperforming the best single SI available. The improvements between the single SI OBMC decoder and the multi-hypothesis decoder ranges from 1.50 to 4.95 dB and from 21.24% to 49.06% Bjøntegaard bit-rate savings.

Sequence	**QP**	**ΔPSNR** [dB]	**ΔRate** [%]
Ballet	L	3.03	−42.80
	20	3.46	−46.62
	30	2.98	−41.53
Breakdancers	L	1.80	−23.95
	20	1.95	−25.55
	30	1.76	−23.37
Dancer	L	4.95	−49.06
	20	4.74	−47.61
	30	4.43	−45.04

Table 3: Bjøntegaard Distances between OBMC and the proposed methods, GOP8.

5. REFERENCES

[1] B. Girod, A.M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71 –83, January 2005.

[2] F. Pereira, "Distributed video coding: basics, main solutions and trends," in *Proc. of IEEE ICME*, Piscataway, NJ, USA, 2009, ICME'09, pp. 1592–1595, IEEE Press.

[3] K. Müller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643 –656, April 2011.

[4] P. Kauff, N. Atzpadin, C. Fehn, M. Müler, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Processing: Image Communication, Special issue on three-dimensional video and television*, vol. 22, no. 2, pp. 217 – 234, 2007.

[5] M. Tanimoto, M.P. Tehrani, T. Fujii, and T. Yendo, "Free-Viewpoint TV," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 67 –76, January 2011.

[6] X. Huang and S. Forchhammer, "Cross-band noise model refinement for transform domain Wyner-Ziv video coding," *Signal Processing: Image Communication*, vol. 27, no. 1, pp. 16 – 30, 2012.

[7] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *EURASIP Signal Processing Journal*, vol. 86, no. 11, pp. 3123–3130, November 2006.

[8] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Motion vector sharing and bitrate allocation for 3D video-plus-depth coding," *EURASIP J. Appl. Signal Process.*, vol. 2009, pp. 1–13, January 2008.

[9] X. Huang, L.L. Rakêt, H.V. Luong, M. Nielsen, F. Lauze, and S. Forchhammer, "Multi-hypothesis transform domain Wyner-Ziv video coding including optical flow," in *Proc. of IEEE MMSP*, October 2011, pp. 1 –6.

[10] H. Oh and Y.-S. Ho, "H.264-based depth map sequence coding using motion information of corresponding texture video," in *Proc. of PSIVT*, Berlin, Heidelberg, 2006, pp. 898–907, Springer-Verlag.

[11] M. Winken, H. Schwarz, and T. Wiegand, "Motion vector inheritance for high efficiency 3D video plus depth coding," in *Proc. of IEEE PCS*, May 2012, pp. 53 –56.

[12] G. Petrazzuoli, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu, "Wyner-Ziv coding for depth maps in multiview video-plus-depth," in *Proc. of IEEE ICIP*, September 2011, pp. 1817 –1820.

[13] M. Salmistraro, M. Zamarin, and S. Forchhammer, "Wyner-Ziv Coding of Depth Maps Exploiting Color Motion Information," *Proceedings of SPIE, the International Society for Optical Engineering*, vol. 8666, pp. 8666–14, 2013.

[14] Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *Image Processing, IEEE Transactions on*, vol. 11, no. 12, pp. 1442 – 1449, December 2002.

[15] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[16] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L^1 optical flow," in *In Ann. Symp. German Association Patt. Recogn*, 2007, pp. 214–223.

[17] L.L. Rakêt, L. Roholm, M. Nielsen, and F. Lauze, "TV-L^1 optical flow for vector valued images," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Yuri Boykov *et al.*, Ed., vol. 6819 of *Lecture Notes in Computer Science*, pp. 329–343. Springer, 2011.

[18] L.L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert, "Motion compensated frame interpolation with a symmetric optical flow constraint," in *Advances in Visual Computing*, George Bebis *et al.*, Ed., vol. 7431 of *Lecture Notes in Computer Science*, pp. 447–457. Springer Berlin Heidelberg, 2012.

[19] B. Goldluecke, E. Strekalovskiy, and D. Cremers, "The natural vectorial total variation which arises from geometric measure theory," *SIAM Journal on Imaging Sciences*, vol. 5, no. 2, pp. 537–563, 2012.

[20] L.C. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, 2004.

[21] "Extension of existing 3DV test set toward synthetic 3D video content," ISO/IEC JTC1/SC29/WG11, Doc. M19221, Daegu, Korea, January 2011.

[22] "DISCOVER project test conditions," December 2007, http://www.img.lx.it.pt/ discover/test_conditions.html.

[23] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," in *VCEG Meeting*, Austin, USA, April 2001.

[24] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: Architecture, techniques and evaluation," *Proc. of IEEE PCS*, November 2007.

Wyner-Ziv Coding of Depth Maps Exploiting Color Motion Information

Matteo Salmistraro, Marco Zamarin, Søren Forchhammer

Department of Photonics Engineering, Technical University of Denmark,
Ørsteds Plads, 2800 Kgs. Lyngby, Denmark

ABSTRACT

Distributed Video Coding of multi-view data and depth maps is an interesting and challenging research field, whose interest is growing thanks to the recent advances in depth estimation and the development of affordable devices able to acquire depth information. In applications like video surveillance and object tracking, the availability of depth data can be beneficial and allow for more accurate processing. In these scenarios, the encoding complexity is typically limited and therefore distributed coding approaches are desirable. In this paper a novel algorithm for distributed compression of depth maps exploiting corresponding color information is proposed. Due to the high correlation of the motion in color and corresponding depth videos, motion information from the decoded color signal can effectively be exploited to generate accurate side information for the depth signal, allowing for higher rate-distortion performance without increasing the delay at the decoder side. The proposed scheme has been evaluated against state-of-the-art distributed video coding techniques applied on depth data. Experimental results show that the proposed algorithm can provide PSNR improvement between 2.18 dB and 3.40 dB on depth data compared to the reference DISCOVER decoder, for GOP 2 and QCIF resolution.

Keywords: Distributed Video Coding, Depth map coding, Motion estimation, Side Information generation, Color plus Depth coding, Motion Vector sharing.

1. INTRODUCTION

Recent advances show that Distributed Video Coding (DVC) is becoming a feasible alternative to traditional video coding in scenarios in which the encoding complexity is limited by strict constraints[1,2]. A typical scenario is multi-camera video surveillance, in which power consumption constraints require low-complexity encoders while complexity constraints are relaxed at the decoder side. In a typical DVC setting[3], a subset of the input frames, called "key frames" (KF), is intra-coded (i.e. coded without exploiting temporal redundancy) and transmitted; for the remaining frames, called "Wyner-Ziv frames", parity check data are calculated and made available for transmission. At the decoder side, intra-coded frames are decoded and missing frames are estimated from the available decoded ones. Estimated frames, referred as side information (SI), are then corrected, if needed, using parity bits from the encoder. The generation of the SI is of crucial importance as it directly impacts the amount of additional data required to the encoder and thus the coding efficiency of the whole scheme. Distributed coding of multi-view video has also been proposed in the literature[1]: due to the high intrinsic redundancy of multi-view data, it is possible to generate SIs for intermediate views allowing for efficient coding performance. A popular and compact 3D video representation is the so-called "video-plus-depth" in which depth map stream(s) are associated to mono-/multi-view video(s)[4]. Depth data allows for easy generation of virtual views through depth-image-based-rendering (DIBR) algorithms[4], making this representation suitable for flexible implementations of 3DTV and free viewpoint TV (FTV)[5]. Moreover, depth data can be exploited for other purposes, such as activity detection, object tracking and background/foreground separation[6], which might be of interest in a video surveillance scenario. Due to the high correlation of the motion in the color and depth videos, color information can be effectively exploited to predict motion in the depth sequence, this basic idea has been exploited in various works[7,8]. The same concept can be exploited in a DVC scenario for accurate SI generation, for example, in the work of Petrazzuoli et al.[9] in which multiple decoded color frames are exploited.

Further author information:
Matteo Salmistraro: E-mail: matsl@fotonik.dtu.dk, Telephone: +45 45256635

M. Salmistraro, M. Zamarin, S. Forchhammer, "Wyner-Ziv Coding of Depth Maps Exploiting Color Motion Information", *Proc. of the 2013 IS&T/SPIE Visual Information Proc. and Comm. Conf. (EI 2013)*, Burlingame, CA, USA, Feb. 3-7, 2013.

This paper addresses Wyner-Ziv (WZ) coding of depth data in a video-plus-depth scenario. In the considered framework color data are available at the decoder and therefore can be exploited in the Wyner-Ziv decoding of depth data. Specifically, motion vectors for color data are used on corresponding depth images allowing improving the performance of state-of-the-art DVC schemes on depth sequences. Differently from the work of Petrazzuoli et al.[9], in the proposed scheme the color frame corresponding to the to-be-decoded depth frame is available at the decoder thus allowing for accurate handling of motion. Moreover, with the proposed approach the decoding delay is not increased, as only the corresponding color frames of the current Group-Of-Pictures (GOP) are required in order to decode the (WZ encoded) depth frames. We investigated the performance for GOPs 2, 4, and 8. The remainder of this paper is organized as follows: in Section 2 the proposed depth SI generation algorithm is described and discussed. Section 3 presents experimental results highlighting the benefits of the proposed method. Finally Section 4 closes the paper and outlines future work.

2. PROPOSED METHOD

The DVC decoder used in this work is based on the one proposed by Huang et al.[10] and it is depicted in Fig. 1. At the encoder the WZ frames are DCT transformed and quantized. The data are organized in bitplanes and parity bits are calculated for each of them by using a channel coder. A subset of the parity bits is sent to the decoder. The decoder tries to correct the errors in the co-located bitplane in the SI; if the decoding does not give an acceptable solution new parity bits are required. The main improvements of this decoder[10] compared with DISCOVER[11] are an improved SI generation system, called Overlapped Block Motion Compensation (OBMC), and an improved noise modelling technique, in which previously decoded DCT coefficients are exploited to refine the noise modelling of the to-be-decoded coefficients.

In this work, the SI generation and the residual estimation modules have been taken as starting point and improved in order to exploit color motion information for depth coding. When decoding the depth frame at instants t (referred as D_t), depth KFs at instant $t-1$ and $t+1$ (called D_{t-1} and D_{t+1}, respectively) can be exploited for SI generation in the case of GOP 2. Coding of depth maps using DVC is challenging as smooth and regular areas may mislead the Motion Estimation (ME) scheme employed by the SI generation module at the decoder. In the scenario considered color frames at instant $t-1$, t and $t+1$ (called C_{t-1}, C_t and C_{t+1}, respectively) are also available at the decoder (see Fig. 2).

If depth and color edges are perfectly aligned, the corresponding Motion Vectors (MVs) present a similar behavior[9]. We exploit this fact in order to effectively predict D_t: first MVs between frames C_t and C_{t-1}, and between C_t and C_{t+1} are calculated. Then, the two motion fields are used to compensate the depth frames D_{t-1} and D_{t+1} thus originating the two depth SIs Y_{t-1} and Y_{t+1}, respectively. Finally, the average $Y=(Y_{t-1}+Y_{t+1})/2$ of these two images is used as SI for the current to-be-decoded depth frame D_t. Together with the SI, an

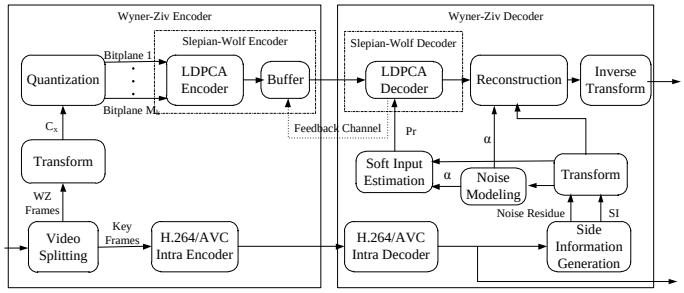


Figure 1: The DVC codec presented in[10].

estimation of the residual signal, i.e. the difference between the original frame D_t and the SI Y, has to be calculated. The residual is important in order to calculate the reliability of the SI: a good estimation is crucial for an efficient parity bit allocation. Since D_t is not accessible at the decoder, the residual has to be estimated. In our case the absolute difference between Y_{t-1} and Y_{t+1} has been used as residual signal. The argument behind this simple but effective choice is that if a depth region in D_t is well predicted in Y, there is a high chance that the corresponding values in Y_{t-1} and Y_{t+1} are similar. Instead, regions of D_t that are not well predicted will most likely present significant differences between Y_{t-1} and Y_{t+1} in the corresponding regions. These type of residuals are commonly referred to as *on-line residuals*, as opposed to *off-line residuals*. An off-line residual-based decoder uses as residual $D_t - Y$. This choice is clearly unrealistic because D_t is not available at the decoder, but it gives an upper bound to the maximum achievable performance of a decoder. The performance of the ME module is also crucial: minor deviations in the color frames could create noticeable depth mismatches at the decoder (e.g. regions having similar texture might belong to objects with different distances from the camera plane), resulting in low quality SIs. After various tests, the so-called "Adaptive Rood Pattern Search" (ARPS) ME algorithm[12] has been adopted. Even though this method might not provide the lowest Mean Squared Error (MSE) between Y and D_t, it is the one which generates fewer artefacts. This algorithm uses an adaptive pattern to search the corresponding blocks in the reference frames, and the search is refined using a two-step refinement procedure. Due to these features the ARPS algorithm outperformed the full search ME algorithm in our scenario and was thus selected for the experiments.

3. EXPERIMENTAL RESULTS

The proposed SI generation method has been evaluated on the Breakdancers and Ballet[13] DIBR sequences (view 4, 100 frames for both color and depth) with QCIF resolution at 15 fps. KFs have been intra-coded by means of H.264/AVC as in[14]. The following QPs have been used for depth data: 31, 34, 37, 40, with the following quantization matrices: Q7, Q6, Q4, Q1[14]. Uncompressed color streams have been used. The proposed method has been compared against the scheme proposed by Huang et al.[10] (denoted as [Huang2012]) and the reference DISCOVER decoder[11], both applied to depth map sequences. The rate-distortion (RD) curves for the Wyner-Ziv frames (GOP 2) are shown in Fig. 3 for both on-line and off-line residual estimations. Results show that the proposed method significantly outperforms the DISCOVER codec and provides a gain also over the OBMC-based decoder[10] for both the test sequences. The presented approach suffers in case of high motion data and when significant occlusions between frames occur, like in the Breakdancers sequence. When an occlusion occurs, blocks in C_t are not matched with blocks in C_{t-1} or C_{t+1} due to overlapping objects. This leads to wrong matching, and when wrongly matched MVs are applied to the depth maps, noticeable artefacts are introduced. Better performance can probably be obtained if specialized ME methods and SI fusion algorithms are considered.

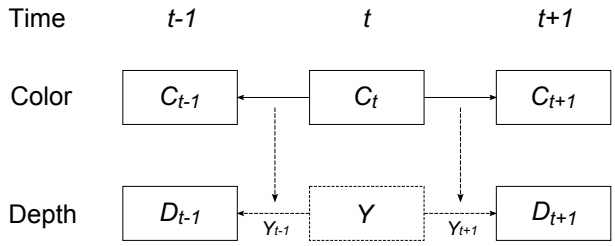
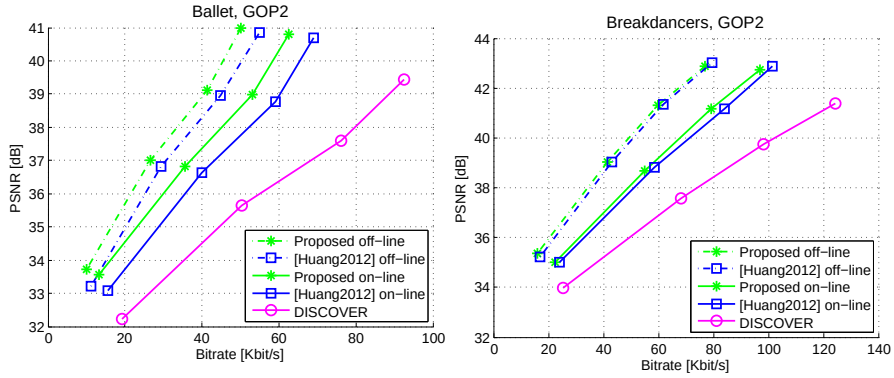


Figure 2: Depth SI generation.



(a) Ballet (b) Breakdancers

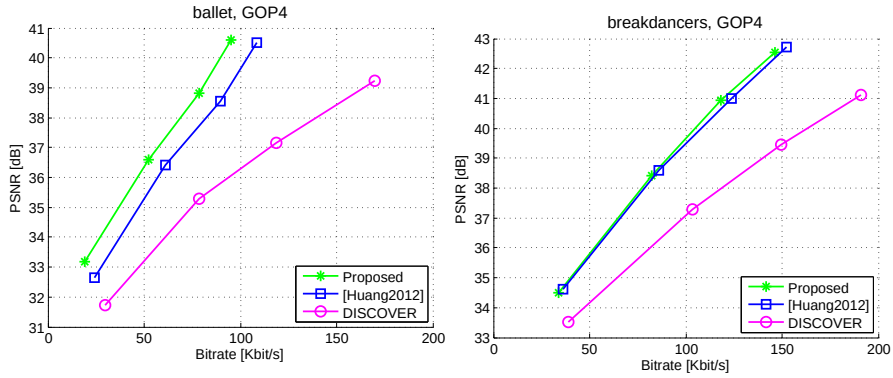
Figure 3: GOP 2, RD performance, WZ frames only, QCIF resolution.

Rate and distortion Bjøntegaard differences[15] have also been evaluated for the proposed method with on-line residual. In the case of Breakdancers the PSNR gain is 0.30 dB and the bit-rate saving is 3.71% when compared to the OBMC-based decoder[10], and 2.18 dB and 35.08% when compared to DISCOVER. As for Ballet, the PSNR gain is 1.62 dB and the bit-rate saving is 24.44% when compared to the OBMC-based decoder[10], and 3.40 dB and 46.39% when compared to DISCOVER. The proposed on-line residual shows acceptable performance as the RD performance of the on-line residuals-based decoders and those of the off-line residuals-based decoders are close. After this validation we have examined the performance of the system for GOPs 4 and 8. In both cases the proposed method outperforms DISCOVER and also outperforms the OBMC-based decoder[10]. In the worst case (see Fig. 5b) the RD performance of the two methods are basically the same, as can be also seen from Table 1, where the Bjøntegaard distances between the proposed method and[10] are summarized.

Having images with QCIF resolution is a quite widespread set-up in DVC coding. Nevertheless, depth maps are usually used at higher resolution, hence we investigated also the CIF scenario. In this case we used the quantization matrices Q8, Q7, Q4, Q1[14] and QPs for the depth data 29, 31, 37, 40. The results for all the GOPs can be seen in Figs. 6 to 8, and are summarized using Bjøntegaard distances in Table 2.

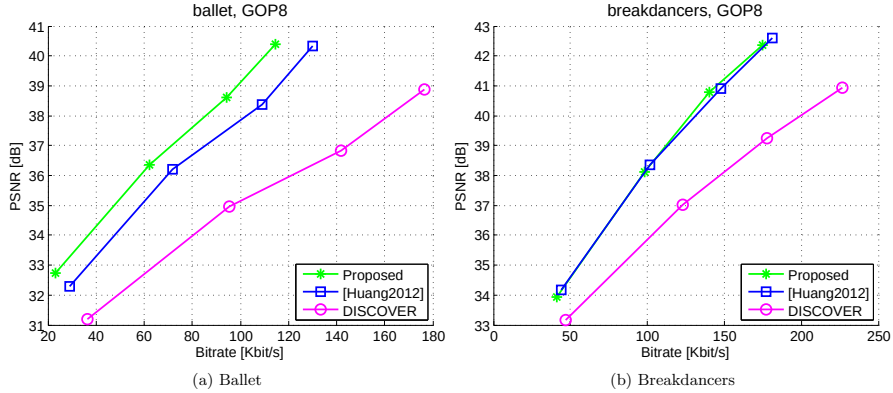
Sequence	**GOP**	**ΔPSNR** [dB]	**ΔRate** [%]
Ballet	2	1.62	−24.44
	4	1.71	−29.06
	8	1.86	−25.51
Breakdancers	2	0.31	−3.71
	4	0.13	−1.97
	8	0.07	−0.53

Table 1: Bjøntegaard Distances between [Huang2012] and the proposed method, WZ frames only, QCIF resolution.



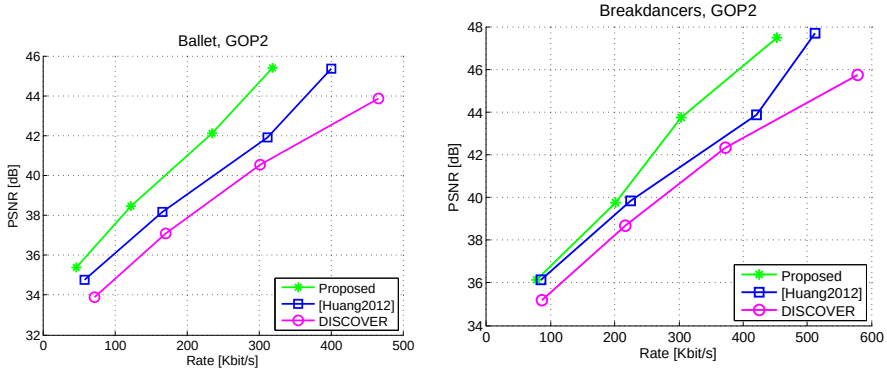
(a) Ballet (b) Breakdancers

Figure 4: GOP 4, RD performance, WZ frames only, QCIF resolution.



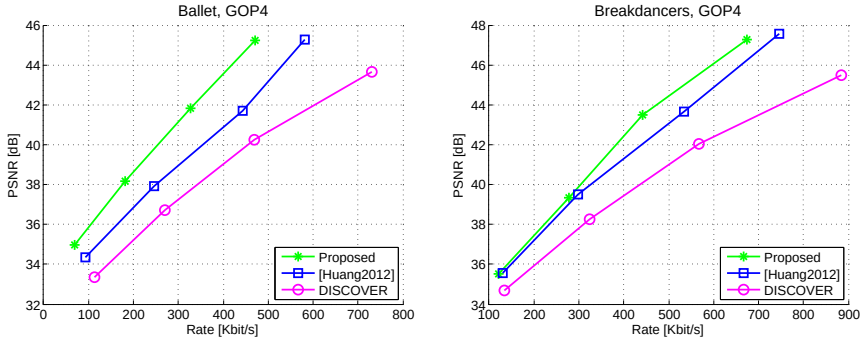
(a) Ballet (b) Breakdancers

Figure 5: GOP 8, RD performance, WZ frames only, QCIF resolution.



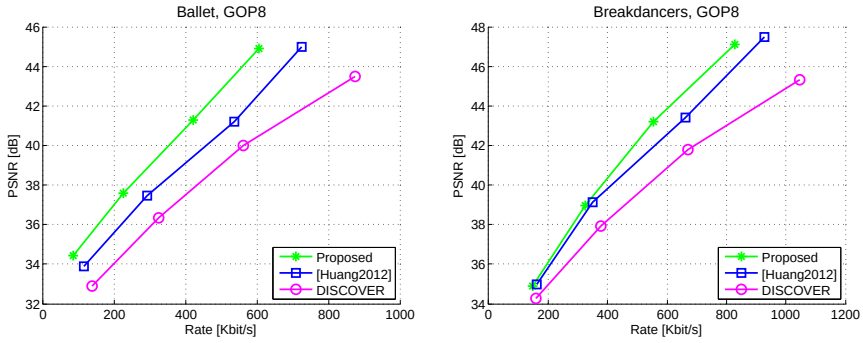
(a) Ballet (b) Breakdancers

Figure 6: GOP 2, RD performance, WZ frames only, CIF resolution.



(a) Ballet (b) Breakdancers

Figure 7: GOP 4, RD performance, WZ frames only, CIF resolution.



(a) Ballet (b) Breakdancers

Figure 8: GOP 8, RD performance, WZ frames only, CIF resolution.

4. CONCLUSIONS

In this work we explored applying a modern DVC decoder to the depth map coding problem exploiting color motion information. The main contribution of the paper is a novel depth SI generation method able to exploit motion information from corresponding decoded color frames. Experimental results show that the proposed algorithm can outperform both the method proposed by Huang et al.[10] and the reference DISCOVER codec. Future work includes multi-hypothesis decoding and development of specialized ME algorithms for more efficient depth motion prediction.

REFERENCES

[1] Guillemot, C., Pereira, F., Torres, L., Ebrahimi, T., Leonardi, R., and Ostermann, J., "Distributed monoview and multiview video coding," *Signal Processing Magazine, IEEE* **24**, 67 –76 (sept. 2007).

[2] Pereira, F., "Distributed video coding: basics, main solutions and trends," in [*Proceedings of the 2009 IEEE international conference on Multimedia and Expo*], *ICME'09*, 1592–1595, IEEE Press, Piscataway, NJ, USA (2009).

[3] Girod, B., Aaron, A., Rane, S., and Rebollo-Monedero, D., "Distributed video coding," *Proceedings of the IEEE* **93**, 71 –83 (jan. 2005).

Sequence	**GOP**	**ΔPSNR** [dB]	**ΔRate** [%]
Ballet	2	1.79	−29.79
	4	1.88	−29.70
	8	1.60	−25.03
Breakdancers	2	0.28	−16.61
	4	0.46	−9.45
	8	0.56	−8.73

Table 2: Bjøntegaard Distances between [Huang2012] and the proposed method, WZ frames only, CIF resolution.

[4] Kauff, P., Atzpadin, N., Fehn, C., Müler, M., Schreer, O., Smolic, A., and Tanger, R., "Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability," *Signal Processing: Image Communication, Special issue on three-dimensional video and television* **22**(2), 217 – 234 (2007).

[5] Tanimoto, M., Tehrani, M., Fujii, T., and Yendo, T., "Free-viewpoint tv," *Signal Processing Magazine, IEEE* **28**, 67 –76 (jan. 2011).

[6] Mehrotra, S., Zhang, Z., Cai, Q., Zhang, C., and Chou, P., "Low-complexity, near-lossless coding of depth maps from kinect-like depth cameras," in [*Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*], 1 –6 (oct. 2011).

[7] Oh, H. and Ho, Y.-S., "H.264-based depth map sequence coding using motion information of corresponding texture video," in [*Proceedings of the First Pacific Rim conference on Advances in Image and Video Technology*], *PSIVT'06*, 898–907, Springer-Verlag, Berlin, Heidelberg (2006).

[8] Winken, M., Schwarz, H., and Wiegand, T., "Motion vector inheritance for high efficiency 3d video plus depth coding," in [*Picture Coding Symposium (PCS), 2012*], 53 –56 (may 2012).

[9] Petrazzuoli, G., Cagnazzo, M., Dufaux, F., and Pesquet-Popescu, B., "Wyner-ziv coding for depth maps in multiview video-plus-depth," in [*Image Processing (ICIP), 2011 18th IEEE International Conference on*], 1817 –1820 (sept. 2011).

[10] Huang, X. and Forchhammer, S., "Cross-band noise model refinement for transform domain wynerziv video coding," *Signal Processing: Image Communication* **27**(1), 16 – 30 (2012).

[11] Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., and Ouaret, M., "The DISCOVER codec: Architecture, Techniques and Evaluation," in [*Picture Coding Symposium (PCS'07)*], (2007).

[12] Nie, Y. and Ma, K.-K., "Adaptive rood pattern search for fast block-matching motion estimation," *Image Processing, IEEE Transactions on* **11**, 1442 – 1449 (dec 2002).

[13] Zitnick, C. L., Kang, S. B., Uyttendaele, M., Winder, S., and Szeliski, R., "High-quality video view interpolation using a layered representation," in [*ACM SIGGRAPH 2004 Papers*], *SIGGRAPH '04*, 600–608, ACM, New York, NY, USA (2004).

[14] "Discover project test conditions," (December 2007). http://www.img.lx.it.pt/ discover/test_conditions.html.

[15] Bjontegaard, G., "Calculation of average psnr differences between rd-curves," in [*VCEG Meeting*], *VCEG Meeting* (April 2001).

2012 IEEE International Conference on Multimedia and Expo Workshops

Lossless Compression of Stereo Disparity Maps for 3D

Marco Zamarin, Søren Forchhammer
Department of Photonics Engineering
Technical University of Denmark
Kongens Lyngby, Denmark
{*mzam, sofo*}*@fotonik.dtu.dk*

***Abstract*—Efficient compression of disparity data is important for accurate view synthesis purposes in multi-view communication systems based on the "texture plus depth" format, including the stereo case. In this paper a novel technique for lossless compression of stereo disparity images is presented. The coding algorithm is based on bit-plane coding, disparity prediction via disparity warping and context-based arithmetic coding exploiting predicted disparity data. Experimental results show that the proposed compression scheme achieves average compression factors of about 48:1 for high resolution disparity maps for stereo pairs and outperforms different standard solutions for lossless still image compression. Moreover, it provides a progressive representation of disparity data as well as a parallelizable structure.**

***Keywords*-Lossless Coding, Stereo Disparity Images, Context-Coding, Disparity Warping, 3DTV**

I. Introduction

Depth and disparity maps are exploited in different coding paradigms for 3D video coding. The availability of one or more depth maps at both encoder and decoder sides allows for advanced processing operations - such as disparity compensation and view synthesis based on the Depth-Image-Based Rendering (DIBR) [1] paradigm - that play a key role in different compression and transmission schemes [2]. Therefore, efficient solutions for depth compression are required. Disparity images typically consist of smooth regions divided by sharp edges. While standard compression schemes for natural images and videos - such as JPEG [3], JPEG2000 [4], and H.264/AVC [5] - can be used, it is clear that optimal compression can be achieved only with coding approaches specifically targeted to depth and disparity images, due to their substantial structural difference compared to natural images.

A number of specialized coding approaches have recently been proposed in the literature for lossy compression of depth data. For instance, Morvan *et al.* introduced in [6] a platelet-based scheme able to preserve sharp edges - crucial for rendering purposes - even at low bit-rates. In [7] depth coding is performed by combining graph-based transform and transform domain sparsification techniques in order to achieve effective compression performance while preserving good view synthesis capabilities. New coding approaches able to exploit the redundancy between color data and corresponding depth data have also recently been proposed, like [8] in which segmented color data are exploited to predict the shape of the different surfaces in the depth map.

Most of these schemes are motivated by the fact that depth data are typically used for view synthesis purposes and are not supposed to be directly displayed, therefore any coding distortion reflecting in a tolerable impact on the synthesis performance can be conveniently exploited to improve compression efficiency. While on one hand this approach allows for very effective solutions, on the other hand it has not been shown yet that lossless mono-/multi-view depth and disparity map compression can be performed very efficiently when specialized schemes are applied.

In some scenarios, highly efficient lossless compression of depth data can provide an attractive alternative to lossy coding - at least at high data rates - since no distortion due to depth coding is introduced on synthesized views. Moreover, depth information can be used for a number of other purposes - including activity detection, object and face tracking, and scene matting - in which lossless or near-lossless coding can be of interest in order to allow for very accurate processing.

Only a few works on efficient lossless compression of mono-view depth data have been proposed in the literature and to the best of our knowledge lossless compression of stereo/multi-view depth data is still an unexplored field. In [9] a bit-plane-based approach using binary shape coding and motion estimation is presented for the case of depth video sequences. In [10] Heo and Ho proposed an improved version of the arithmetic encoder of H.264/AVC specifically targeted to depth data able to provide an average bit-rate reduction of 4% compared to the standard H.264/AVC arithmetic encoder. In [11] lossless compression is performed exploiting minimum description length segmentation. Finally, Mehrotra *et al.* proposed in [12] a low-complexity near-lossless coding scheme for depth images from range cameras based on depth-precision-based quantization and entropy coding.

In this paper we propose a novel, high-efficiency lossless coding scheme for stereo disparity pairs. While the left map is intra-coded by means of bit-plane decomposition and context-based arithmetic coding, the right map is predicted from the left one via disparity warping - similarly to the "depth synthesis prediction" step in [13] - and encoded by

978-0-7695-4729-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICMEW.2012.113

M. Zamarin, S. Forchhammer, "Lossless Compression of Stereo Disparity Maps for 3D", *Proc. of the 2012 IEEE Int'l Work. on Hot Topics in 3D Multimedia (Hot3D 2012)*, pp. 617-622, Melbourne, Australia, July 13, 2012.

means of context-coding including pixels from the predicted image in the template. Due to the very high correlation between the left and right maps, the latter can be effectively encoded at very low bit-rates. Moreover, with such approach it is possible to progressively decode only the left disparity map, in case a single disparity map is needed. Once the left map has been decoded, the right one can also be decoded in a progressive way.

The remainder of this paper is organized as follows. In Section II the proposed coding scheme is introduced and analyzed. In Section III the coding performance of the proposed approach are shown and compared with the ones of different state-of-the-art lossless codecs for still images, highlighting the advantages of the proposed scheme. Finally, Section IV concludes the paper.

II. Proposed coding scheme

As mentioned in Section I, the proposed coding scheme includes two main steps: first, the left disparity map is intra-coded, then the right one is predicted via disparity warping and encoded exploiting this prediction. Consequently, the produced bit stream is split into two parts: one related to the left map (called "left" bit stream) and one related to the right one (called "right" bit stream). The coding steps are described in detail in the following subsections, starting with the description of the warping algorithm.

A. Disparity map warping

In our experiments we considered 8-bit disparity stereo images from the Middlebury Repository [14], [15]. These disparity maps are rectified and radial distortion has been removed. Therefore, corresponding pixels in the left and right maps are horizontally shifted, and the shifting lengths are directly specified by the pixel values. In such a scenario, a simple but effective algorithm to warp the left disparity map to the viewpoint of the right one horizontally shifts single pixels according to their value. The warping process is described by Algorithm 1, in which l and w are the left and warped disparity maps, respectively, and x and y are the column and row coordinates of the pixels in the maps, respectively ($1 \leq x \leq W$, $1 \leq y \leq H$).

Note that while the order in which the rows are browsed is completely irrelevant, the columns need to be visited from the leftmost one ($x = 1$) to the rightmost one ($x = W$) in order to properly handle objects overlapping.

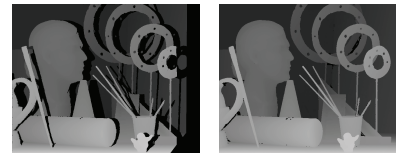
Due to the granularity of the shifting lengths in the warping process, some one-pixel-thick lines may appear in the warped disparity map. However, pixels of those lines are easy to detect and recover by means of median filtering. Specifically, a pixel in a one-pixel-thick line is recovered by the median of the 3×3 pixel window centered in it, excluding from the window pixels in the line.

Figure 1 shows an example of the effectiveness of the proposed warping algorithm. Black pixels in Fig. 1(a) and (b) refer to unknown disparity values. While in the original map (b) black pixels only appear due to uncertainties in the original data, in the warped map (a) black holes appear for two reasons: as an effect of the holes in the original map and due to disoccluded regions, i.e. regions of the warped map that are not visible in the original map because of objects overlapping.

Algorithm 1: Disparity map warping

Input : Left disparity map l
Output: Warped left disparity map w

```
for y ← 1 to H do
    for x ← 1 to W do
        if l(x, y) > 0 then
            v ← x − l(x, y)
            if v ≥ 1 then
                w(v, y) ← l(x, y)
            end
        end
    end
end
```



(a) Warped left map (b) Right map

Figure 1. Example of disparity map warping of the left map of the "Art" dataset [14] (a) and original right map (b). Black pixels indicate unknown disparity values.

B. Intra coding

The lossless intra coding of the left disparity map is performed in three steps, as shown in Fig. 2(a) (upper part):

1) **Bit-planes decomposition** – The disparity map is decomposed in its n bit-planes $B_i, i = 1, 2, \ldots, n$, B_1 being the most-significant bit-plane and B_n the least-significant one (typical values of n are 8 and 16).
2) **Gray coding** – The bit-planes are Gray-coded in order to obtain a higher local correlation of the bits within each bit-plane, as done in [9]. This operation corresponds to the calculation of the *exclusive or* among adjacent bit-planes: $B_1^{\text{Gray}} = B_1; B_i^{\text{Gray}} = B_i \oplus B_{i-1}, i = 2, 3, \ldots, n$.
3) **Context-based arithmetic coding** – The Gray-coded bit-planes are encoded by means of context-based arithmetic coding in a JBIG [16] -like manner from

B_1^{Gray} to B_n^{Gray}. In the i-th bit-plane, template pixels are chosen among the ones in the causal (black) 30-pixel search area of Fig. 3 from the same bit-plane, and from the set of $i-1$ pixels defined by pixels at the same position of the one being encoded placed in an already encoded B_j^{Gray} bit-plane, $1 \leq j < i$. Pixels are chosen in a greedy fashion: the template is populated by choosing at every iteration the pixel in the search area that minimizes the ideal code length needed to encode the current bit-plane. The template growing stops when any new pixel does not reduce the ideal code length anymore. Arithmetic coding is then applied using the obtained template.

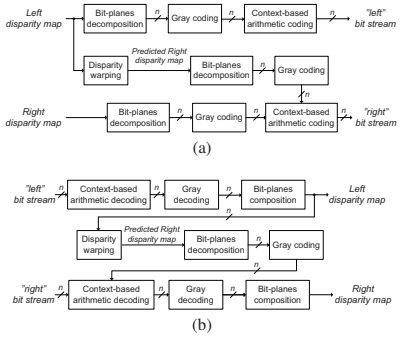


Figure 2. Proposed encoder (a) and decoder (b) block schemes.

The structure of the encoder ensures a progressive representation of the Intra-coded map, which can be of interest in many applications.

C. Inter coding

The encoding of the right disparity map follows the same steps, with the difference that in context-coding pixels from a predicted version of the right map are used, as shown in Fig. 2(a).

The right disparity map is predicted from the left one via disparity warping as described in the Subsection II-A. The predicted map is then decomposed in bit-planes and Gray-coded. Due to the very high correlation between the right disparity map and its prediction from the left map (see Fig. 1), pixels in the predicted map can be effectively used in the context-coding step to significantly improve the compression performance. Specifically, a new search area including pixels from both the current bit-plane being encoded and its corresponding bit-plane in the predicted map can be exploited. Moreover, since the left disparity map has already been encoded and will therefore be available

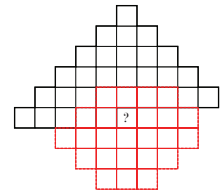


Figure 3. The 30-pixel "local" causal search area (black) and the 25-pixel "predicted" non-causal search area (dashed red). "?" indicates the current pixel being encoded.

at the decoder side, a non-causal search area can be chosen in the predicted bit-plane, including the pixel at the same position of the one being encoded. In this way, statistics on the "future" behavior of the image are introduced in the context-coding process with a significant impact on its performance. Figure 3 shows both the 30-pixel search area from the current bit-plane (black) and the 25-pixel search area from the predicted bit-plane (dashed red). The overall size of the available search area for a pixel in the i-th bit-plane is $55+(i-1)$ pixels, since pixels in previously encoded bit-planes at the same position of the one being encoded are also considered, as in the Intra case. The template is populated following the same strategy used in the Intra case.

Thanks to the bit-plane-based structure of the Inter-coding method, the right disparity map is also encoded in a progressive manner. Moreover, since the bit-planes are encoded separately, parallel encoding can easily be applied to both the Intra and Inter schemes.

D. Decoding process

The decoding is performed by simply reversing the encoding process. Firstly, the left disparity map is decoded by means of arithmetic decoding, Gray decoding and bit-plane composition. Then, in case the right disparity map is also needed, the decoded left disparity map is warped with the warping algorithm described in Subsection II-A, and the warped image is exploited in the context-based arithmetic decoding of the right map compressed bit stream. Finally, Gray decoding and bit-plane composition allow to reconstruct the original right map. The complete decoder block scheme is shown in Fig. 2(b).

III. Experimental Results

A. Test images

The proposed algorithm has been evaluated on 8-bit stereo disparity images from the "2005" and "2006" datasets of the Middlebury Repository [14]. These datasets include disparity images in three different resolutions, namely "full" ($\sim$1390$\times$1110), "half" ($\sim$695$\times$555), and "third" ($\sim$463$\times$370). Since disparity values in all of them refer to

the "full" resolution setting, disparity scaling is needed when "half" or "third" resolution images are used for warping purposes. In this case line 4 of Algorithm 1 needs to be updated with the following one: $v \leftarrow x - \langle l(x,y)/d \rangle$, $\langle \cdot \rangle$ being the rounding operator and d a scaling factor equal e.g. to 2 or 3 in case of "half" or "third" resolution images, respectively. Performance have been evaluated in terms of *compression factor*, defined as the ratio between the uncompressed size of a disparity image and its corresponding compressed size. Since the proposed algorithm has been tested on 8-bit disparity maps, the uncompressed size (in bits) of a map is simply given by its total number of pixels multiplied by 8. The compression factor of a stereo pair is defined in the same way.

B. Performances of the Intra and Inter methods

In order to evaluate the performance of the proposed algorithm two Tables are presented. A performance comparison of the proposed Intra and Inter coding methods is provided in Table I in order to show the effectiveness of the Inter method over the Intra one. The compression factors refer to the right disparity maps of the specified datasets. In case of Inter coding, prediction data have been obtained from the corresponding left map. As can be noticed, the Inter method significantly reduces the average bit-rate compared to the Intra case at all the resolutions. The average bit-rate savings over the Intra case are 29.3% for low resolution maps, 39.5% for medium resolution maps, and 52.2% in case of high resolution maps.

Resolution	Dataset	Intra	Inter
Third	Art	12.044	17.277
	Dolls	8.952	14.875
	Lampshade	17.230	27.403
	Moebius	12.597	20.386
	Plastic	23.924	37.187
	Reindeer	12.950	21.049
	Average	**14.616**	**23.030**
Half	Art	16.774	25.451
	Dolls	12.436	21.898
	Lampshade	22.958	41.648
	Moebius	17.190	28.878
	Plastic	30.959	59.444
	Reindeer	17.415	30.861
	Average	**19.622**	**34.698**
Full	Art	30.377	52.952
	Dolls	22.670	42.725
	Lampshade	41.741	84.186
	Moebius	31.408	57.351
	Plastic	58.029	139.255
	Reindeer	31.287	67.401
	Average	**35.919**	**73.978**

Table I
COMPARISON OF COMPRESSION FACTORS OF RIGHT DISPARITY MAPS AT DIFFERENT RESOLUTIONS FOR THE PROPOSED INTRA AND INTER CODING METHODS.

C. Evaluation against standard lossless codecs

Table II compares the compression performance of the proposed method with standard codecs for lossless image compression. Comparisons have been made against CALIC [17], H.264/AVC lossless (JM 18.0 Reference Software[1] has been used), JPEG2000 [4] lossless and JPEG-LS [18]. Lossless compression based on JBIG has also been considered for comparisons: the image bit-planes have been encoded with JBIG after a Gray-coding step (as done in the proposed Intra method, see subsection II-B). Two different settings have been considered for the H.264/AVC codec: in the first one both the left and right maps have been Intra-coded (column "H.264 II" in Table II), while in the second one the right maps have been encoded by means of motion prediction (column "H.264 IP"). Both the configurations use the "High 4:4:4" profile with 4:0:0 YUV format, high-complexity rate-distortion optimization, CABAC entropy coder and adaptive block size. In the second setting motion estimation is performed with a search range of ± 32 pixels.

While the compression factors of the proposed method refer to the case in which the left disparity map is Intra-coded and the right one is Inter-coded, the compression factors of the other methods refer to the case in which both the maps are independently coded, except for the "IP" configuration of H.264/AVC in which motion estimation is used. The compression factors of the proposed scheme also includes some header bits introduced to signal to the decoder which template pixels to use in the decoding process of each bit-plane.

As highlighted in Table II, the proposed method outperforms all the considered standard codecs for lossless image compression at all the resolutions. As expected, better performance are obtained at high resolution due to the higher local correlation within each bit-plane and the more accurate prediction data generated by the warping algorithm. JPEG2000 appears on average as the least efficient codec for disparity data, but it does provide scalability features not provided by the other methods. JPEG-LS does provide better performance than JPEG2000, but these codecs both perform significantly worse than H.264/AVC, which provides - already with the Intra-only setting - average bit-rate reductions of around 36% to 46% compared to JPEG-LS, depending on the resolution. The H.264/AVC codec achieves better performance when motion prediction is performed. However, the improvement gap is not significant enough to motivate the increase of complexity due to motion estimation, especially at low and medium resolutions. The reasons

[1] Available at http://iphome.hhi.de/suehring/tml

Resolution	Dataset	Proposed	CALIC	JBIG	H.264 IP	H.264 II	JPEG-LS	JPEG2000
Third	Art	14.263	11.342	11.657	10.618	10.022	6.353	4.428
	Dolls	11.084	8.462	8.475	8.442	8.038	5.886	4.282
	Lampshade	21.466	15.925	16.019	16.262	14.387	8.890	7.085
	Moebius	15.534	12.146	12.091	11.998	10.753	7.988	5.480
	Plastic	29.077	23.606	22.346	20.522	18.902	9.757	10.637
	Reindeer	15.704	11.736	12.132	12.085	11.082	7.546	5.687
	Average	**17.855**	**13.870**	**13.787**	**13.321**	**12.197**	**7.737**	**6.267**
Half	Art	20.203	15.661	15.988	13.638	13.023	8.149	5.647
	Dolls	15.643	11.465	11.648	11.536	9.718	7.248	5.211
	Lampshade	29.992	20.821	21.259	20.318	17.922	10.669	8.668
	Moebius	21.519	16.560	16.448	15.477	13.996	9.993	6.761
	Plastic	40.678	30.145	28.862	26.541	23.974	11.635	12.553
	Reindeer	21.549	15.621	16.031	15.467	14.153	8.819	6.934
	Average	**24.931**	**18.379**	**18.373**	**17.163**	**15.464**	**9.419**	**7.629**
Full	Art	38.314	27.621	26.997	21.833	21.835	12.208	8.731
	Dolls	28.830	19.994	19.383	18.767	15.553	10.301	7.325
	Lampshade	55.941	37.612	37.256	29.635	28.979	16.117	13.255
	Moebius	40.300	28.937	27.894	22.760	22.769	14.508	9.816
	Plastic	82.194	57.015	52.697	41.118	40.699	16.831	18.331
	Reindeer	40.497	28.073	27.400	22.738	22.741	12.656	10.534
	Average	**47.679**	**33.209**	**31.938**	**26.142**	**25.429**	**13.770**	**11.332**

Table II
COMPARISON OF COMPRESSION FACTORS ON STEREO DISPARITY MAPS FOR THE PROPOSED METHOD AGAINST STANDARD CODECS FOR LOSSLESS COMPRESSION.

for such a small average improvement are two: first, the H.264/AVC architecture has been developed and optimized for efficient lossy coding of video data and therefore lossless performance have not been considered as a main focus; second, it has been shown in the literature [19] that block-based motion estimation does not represent an optimal way to handle changes due to disparity.

The CALIC coder performs on average slightly better than bit-plane-based JBIG at all the resolutions, and both perform better than H.264/AVC. However, one advantage of bit-plane-based JBIG over CALIC is the ability to provide a progressive data representation. The proposed Intra method operates in a bit-plane-based JBIG-like manner, with the only difference that template pixels are not fixed and can be chosen among a large search area, which eventually includes pixels from previously-encoded bit-planes. A comparison between the proposed Intra method and bit-plane-based JBIG can be made by comparing the "Intra" column of Table I with the "JBIG" column of Table II: due to the possibility to explore a large search area, the proposed Intra method provides higher compression factors than the JBIG-based one at all the resolutions (note that the comparison is not made on the same data: Table I only reports compression factors of one disparity map per dataset while Table II provides overall compression factors of stereo pairs. However, Intra performance do not change significantly between the left and right map of a stereo pair). Moreover, as can be noticed from a similar comparison, the proposed Intra method not only outperforms the bit-plane-based JBIG method but also the CALIC codec. By introducing Inter-coding, the proposed method further increases the performance gap with CALIC achieving average bit-rate reductions (over CALIC) of about 22% at low resolution, 26% at medium resolution, and 30% at high resolution.

The average number of template pixels in each Intra-coded bit-plane is 10, with a maximum template size of 13 pixels. As for Inter-coding, the average template size is again 10, but with a maximum size of 16 pixels. The four pixels directly surrounding the current pixel in the "local" search area (see Fig. 3) are frequently selected in both Intra and Inter cases. As for the "predicted" area - as expected - the pixel at the same position as the one being encoded is always selected, very often together with the pixels directly at its left, right, and top. Since a relatively small number of pixels is enough to minimize the bit stream size, the sizes of the "local" and "predicted" search areas can effectively be reduced in order to decrease the algorithm complexity without compromising its efficiency.

Figure 4 shows the disparity maps in which the proposed coder achieves the lowest (Fig. 4(a)) and highest (Fig. 4(b)) compression factors. The geometry described in Fig. 4(a) is clearly more complex than the one described in Fig. 4(b),

but a key role is played by the number of isolated black pixels (which correspond to unknown disparity values) in Fig. 4(a): the presence of such discontinuities affects in a negative way the performance of context-based arithmetic coding.

(a) (b)

Figure 4. "Dolls" (a) and "Plastic" (b) test images from [14].

IV. Conclusion

This paper introduces a novel technique for lossless compression of stereo disparity maps. The coding algorithm firstly encodes the left disparity map (Intra coding) by means of bit-plane decomposition, Gray coding, and context-based arithmetic coding; then encodes the right disparity map in a similar manner exploiting predicted disparity data from the left map (Inter coding). Experimental results show that the proposed Inter method provides average bit-rate reductions between 29.3% and 52.2% over the Intra case, depending on the map resolution. Overall, the proposed scheme achieves average compression factors of about 18:1 for low resolution stereo disparity maps, 25:1 for medium resolution maps, and 48:1 for high resolution maps. The proposed coding scheme is able to outperform the most popular standard solutions for lossless image compression, including JPEG-LS, JPEG2000 lossless and H.264/AVC lossless. Moreover, it provides a progressive representation of disparity data with the possibility to decode only the first disparity map as well as a parallelizable structure.

The algorithm introduced in this paper demonstrates that in the case of depth data, optimal lossless compression can not be obtained with standard coding approaches since these rely on features and structures of natural images that do not appear in depth and disparity maps. Instead, when specialized schemes are adopted, much higher performance can be achieved. Lossless compression factors of about 50:1 can motivate for lossless transmission of depth data in a "texture plus depth"-based communication system thus avoiding rendering artifacts due to lossy disparity coding.

Future developments include better handling of disoccluded regions in the warped disparity map in order to reduce the effect of disparity holes on context-coding, extensions to stereo disparity video, near-lossless coding driven by view synthesis distortion for higher compression gains, and joint texture-disparity coding.

References

[1] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Image Commun.*, vol. 22, pp. 217–234, Feb. 2007.

[2] K. Müller, P. Merkle, and T. Wiegand, "3-D Video Representation Using Depth Maps," *Proc. of the IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[3] G.K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, Feb. 1992.

[4] D.S. Taubman and M.W. Marcellin, *JPEG2000: image compression fundamentals, standards, and practice*, Kluwer Academic Publishers, Boston, 2002.

[5] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.

[6] Y. Morvan, D. Farin, and P.H.N. de With, "Depth-Image Compression based on an R-D Optimized Quadtree Decomposition for the Transmission of Multiview Images," in *IEEE ICIP 2007*, 2007, pp. 205–208.

[7] G. Cheung, A. Ortega, J. Ishida, and A. Kubota, "Depth Map Coding using Graph Based Transform and Transform Domain Sparsification," in *IEEE MMSP 2011*, 2011.

[8] S. Milani, P. Zanuttigh, M. Zamarin, and S. Forchhammer, "Efficient depth map compression exploiting segmented color data," in *IEEE ICME 2011*, July 2011, pp. 1–6.

[9] K.Y. Kim, G.H. Park, and D.Y. Suh, "Bit-plane-based lossless depth-map coding," *Optical Eng.*, vol. 49, no. 6, June 2010.

[10] J. Heo and Y.-S. Ho, "Improved Context-Based Adaptive Binary Arithmetic Coding over H.264/AVC for Lossless Depth Map Coding," *IEEE Signal Process. Letters*, vol. 17, no. 10, pp. 835–838, Oct. 2010.

[11] I. Şchiopu and I. Tăbuş, "MDL Segmentation and Lossless Compression of Depth Images," in *4th Work. on Inf. Th. Meth. in Sc. and Eng. (WITMSE 2011)*, Helsinki, Finland, Aug. 2011.

[12] S. Mehrotra, Z. Zhang, Q. Cai, C. Zhang, and P.A. Chou, "Low-Complexity, Near-Lossless Coding of Depth Maps from Kinect-Like Depth Cameras," in *IEEE MMSP 2011*, 2011.

[13] C. Lee, B. Choi, and Y.-S. Ho, "Efficient multiview depth video coding using depth synthesis prediction," *Optical Eng.*, vol. 50, no. 7, 2011.

[14] "Repository vision.middlebury.edu: Stereo datasets," http://vision.middlebury.edu/stereo.

[15] H. Hirschmuller and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching," *IEEE CVPR 2007*, pp. 1–8, 2007.

[16] JBIG, "Progressive Bi-level Image Compression," in *International Standard ISO/IEC 11544, ITU-T Rec. T.82*, 1993.

[17] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[18] JPEG-LS, "Lossless and near-lossless compression of continuous-tone still images," in *Int'l Standard ISO/IEC 14495*, 1999.

[19] M. Kaaniche, W. Miled, B. Pesquet-Popescu, A. Benazza-Benyahia, and J.-C. Pesquet, "Dense disparity map representations for stereo image coding," in *IEEE ICIP 2009*, Nov. 2009, pp. 725–728.

EFFICIENT DEPTH MAP COMPRESSION EXPLOITING SEGMENTED COLOR DATA

Simone Milani, Pietro Zanuttigh

University of Padova
Department of Information Engineering
Via Gradenigo 6B, 35131 Padova, Italy
{simone.milani, zanuttigh}@dei.unipd.it

Marco Zamarin, Søren Forchhammer

Technical University of Denmark
Department of Photonics Engineering
Ørsteds Plads B343, 2800 Kgs. Lyngby, Denmark
{mzam, sofo}@fotonik.dtu.dk

ABSTRACT

3D video representations usually associate to each view a depth map with the corresponding geometric information. Many compression schemes have been proposed for multi-view video and for depth data, but the exploitation of the correlation between the two representations to enhance compression performances is still an open research issue. This paper presents a novel compression scheme that exploits a segmentation of the color data to predict the shape of the different surfaces in the depth map. Then each segment is approximated with a parameterized plane. In case the approximation is sufficiently accurate for the target bit rate, the surface coefficients are compressed and transmitted. Otherwise, the region is coded using a standard H.264/AVC Intra coder. Experimental results show that the proposed scheme permits to outperform the standard H.264/AVC Intra codec on depth data and can be effectively included into multi-view plus depth compression schemes.

Index Terms— depth map coding, image segmentation, H.264/AVC

1. INTRODUCTION

Differently from stereoscopic visualization, which requires only a couple of views, novel 3D video browsing systems, like free viewpoint video (FVV) architectures and autostereoscopic displays, need an accurate representation of the geometry of the scene. This representation is usually provided by depth maps that need to be compressed and transmitted together with the associated color frames. While many different solutions have been proposed for the compression of standard and multi-view videos, compression of depth data is an open research issue that is attaining a growing interest.

Even if depth information can be compressed by standard image or video coding tools, *ad-hoc* solutions can obtain a higher compression efficiency. Depth maps are usually made of smooth regions (which can be easily approximated by a reduced set of samples) divided by sharp edges. From this premise, the approach by Oh *et al.* [1] subsamples the depth information to obtain higher compression gains and reconstructs it with an adaptive interpolating filter. However, an effective coding requires a careful handling of edge regions. To this purpose, the approach in [2] employs platelets with support areas adapted to the object boundaries. Other approaches resort to a preliminary segmentation of the input color or depth image in order to decompose objects into sets of homogeneous regions. Among these strategies, it is possible to mention the solution by Zhu *et al.* [3] which segments the input frame and adopts a region-based coding for both depth and color signals. The solution in [4] adopts a segmentation of the depth image followed by a linear prediction, while the decomposition into progressive silhouettes in [5] permits obtaining a scalable representation. Note that while segmentation based compression of color data has always been a difficult task [6], the peculiar characteristics of depth maps make this kind of approaches much more effective.

Excluding few recent papers like [7], most of the proposed schemes for multi-view plus depth representation handle the compression of the two data streams as separate tasks. Nevertheless, this work introduces a novel compression scheme for depth images that exploits the corresponding color (texture) image as side information to improve the coding gain in the compression of geometry (depth) data. Segmented data coming from the associated color stream is used to divide the depth map into a set of regions that are assumed to correspond to planar surfaces in the scene. As a matter of fact, the depth values in each segment can be roughly approximated with a plane. If the approximation is accurate enough plane coefficients can be coded and transmitted, otherwise the segment is coded by a standard solution based on H.264/AVC Intra. Finally, we will show how the proposed approach can be efficiently included into a multi-view plus depth coding scheme.

The rest of the paper is organized in the following way: Section 2 presents the proposed coding algorithm, Section 3 discusses the experimental results, and finally, Section 4 draws the conclusions.

978-1-61284-350-6/11/$26.00 ©2011 IEEE

S. Milani, P. Zanuttigh, M. Zamarin, S. Forchhammer, “Efficient Depth Map Compression Exploiting Segmented Color Data”, *Proc. of the 2011 IEEE Int’l Conf. on Multimedia and Expo (ICME 2011)*, pp. 1-6, Barcelona, Spain, July 11-15, 2011.

2. PROPOSED DEPTH CODING ALGORITHM

The proposed coding scheme aims at encoding the depth map by exploiting the side information coming from a segmentation of the associated color image. Fig. 1 reports a block diagram of this architecture, which can be divided into three main steps. In the first step the color image is compressed, and then, the reconstructed color image (which is the only version of color data that will be available at decoder side) is segmented with an ad-hoc procedure (Section 2.1). A surface fitting algorithm is used in the second step to estimate the best planar approximation for each segmented region (Section 2.2). Finally, the plane coefficients are coded in the bit stream whenever the estimated planar surface approximates the current segment accurately enough. In case the planar approximation proves to be inefficient, the region is coded using the H.264/AVC Intra coder (Section 2.3).

The decoding procedure is shown in Fig. 2. At first, the color image is decoded and segmented with the same algorithm used at encoder side. Note how by performing segmentation again at the decoder side we avoid to transmit segmentation data, that would have greatly increased the required bitrate. Then, depth values in each segment are reconstructed from the surface coefficients or from the H.264/AVC bit stream according to the coding mode of each region. Finally in Section 2.4 the proposed scheme is also used into the multi-view plus depth coding scheme of [8].

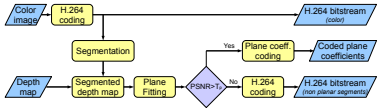


Fig. 1. Encoder block scheme

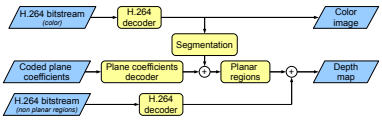
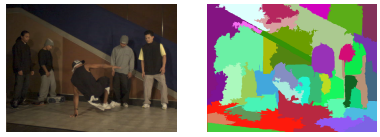


Fig. 2. Decoder block scheme

2.1. Construction of the segmented regions

The first step of the coding routine for depth images consists in partitioning the input depth map into a set of segments that can be easily approximated by a plane. Depth maps are usually made of smooth regions corresponding to the various surfaces in the scenes divided by sharp edges. As a matter of fact, it is possible to assume that each segment can be approximated with a parameterized geometric surface. It is possible



(a) (b)

Fig. 3. Segmented color image for the first frame of the sequence `breakdancers` ($QP = 28$). a) Reconstructed color image; b) Segmentation

to infer this partitioning by applying a segmentation strategy on the coded view. In our tests, we considered different algorithms since the performance of the segmentation strategy significantly affects the coding results. We chose the graph-based segmentation method proposed in [9] since it permits partitioning the input image accurately enough. In our implementation, we set the configuration parameter of the segmentation algorithm in order to generate approximately 100 segments. The smoothing filter has a standard deviation set to half of the standard deviation of the image, and the minimum size of segments is $1/200$ of the image size.

The segmentation algorithm partitions the input image I into N_R pixel regions $R_k \in \mathbb{Z}^2$, and for each R_k the corresponding bounding box is computed. Let us denote with (m_x^k, m_y^k) the coordinates of the upper left corner of the bounding box of R_k and with (M_x^k, M_y^k) the coordinates of the lower right one. Each region is then included in a set of rectangular pixel windows $W_{u,v}^k$ with size $S_k \times S_k$, i.e.

$$\begin{aligned} R_k \subset & \bigcup_{u=0}^{N_{k,x}-1} \bigcup_{v=0}^{N_{k,y}-1} W_{u,v}^k \\ W_{u,v}^k = \{(x,y) \in \mathbb{Z}^2 | & \\ & m_x^k + (u \cdot S_k) \leq x \leq m_x^k + ((u+1) \cdot S_k), \\ & m_y^k + (v \cdot S_k) \leq y \leq m_y^k + ((v+1) \cdot S_k)\} \end{aligned} \tag{1}$$

where

$$\begin{aligned} N_{k,x} &= \lceil (M_x^k - m_x^k)/S_k \rceil \\ N_{k,y} &= \lceil (M_y^k - m_y^k)/S_k \rceil \end{aligned} \tag{2}$$

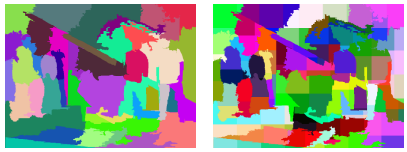
where $S_k \in \mathbb{N}$.

The input depth map is then partitioned into the segments R'_t, which can be defined as

$$\begin{aligned} &\exists\, u, v \qquad 0 \leq u \leq N_{k,x} - 1,\ 0 \leq v \leq N_{k,y} - 1 \\ &R'_t = R_k \cap W_{u,v}^k. \end{aligned} \tag{3}$$

Fig. 4 shows an example of the R'_t regions produced by this additional partitioning. The regions R'_t prove to be smaller than R_k (their maximum size is $S_k \times S_k$), and therefore, it is possible for the fitting strategy described in Section 2.2 to limit the number of outlier points. This fact permits to

improve the performance of the surface fitting. The optimal value of S_k depends on the characteristics of the region R_k to be coded and to the image resolution, for example the `breakdancers` image of Fig. 4 has been coded using $S_k = 128$. In case R_k is accurately fitted by a plane, it is possible to increase the value of S_k. On the other hand, S_k can be reduced for regions presenting non-planar depth signals.



(a) (b)

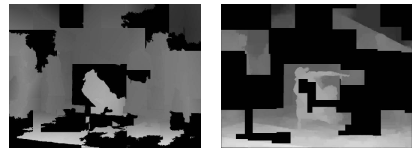
Fig. 4. Construction of the R'_t regions: a) R_k regions produced by the segmentation of the color image (the figure shows a sample frame from the `breakdancers` sequence); b) Corresponding R'_t regions.

2.2. Surface fitting over the segmented regions

After the segmentation step the depth data is divided into a set of regions corresponding to the various objects of the scene. Since most real world objects have planar surfaces, it is reasonable to approximate the regions via a plane fitting of the depth values of pixels in the segment. The proposed scheme is based on the RANSAC [10] algorithm: a set of points is randomly chosen in R'_t, and a linear regression strategy computes the plane that presents the best fit. Then, the distance between each depth sample and the corresponding position on the plane is calculated, and the number of *inliers* (samples whose distance from the plane is less than a threshold T_i) is computed. The procedure is repeated until the MSE value M_t between the estimated plane and the depth data falls below a pre-defined threshold or a predefined maximum number of iterations is reached (this ensures that the algorithm does not loop forever in case of non-planar regions). Finally, the estimate with the highest number of inliers is selected, and the MSE value M_t is used to select the optimal coding strategy in the next stage.

2.3. Coding of the depth data

The RANSAC algorithm computes the best fitting plane for the pixels in each segment R'_t as described in the previous subsection. However, the effectiveness of this approximation depends on the characteristics of the coded depth segment since some parts of the depth map present non-stationary variations and cannot be modeled by a planar approximation. As a matter of fact, these regions can be coded using a standard H.264/AVC coding strategy. Fig. 5 shows an example of the regions coded by the two different strategies.



a) b)

Fig. 5. a) Encoded image by means of plane fitting; b) H.264 encoding of the remaining regions.

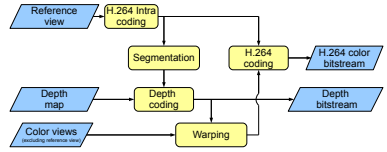


Fig. 6. Proposed multi-view encoding scheme.

After estimating the fitting plane for the current segment R'_t, the PSNR value for pixels in R'_t is computed from the MSE value M_t stored in the previous stage. In case the value is lower than a fixed threshold T_p (which depends on the chosen quantization parameter QP), the algorithm includes R'_t into a block B'_t of size $(N'_{t,x} \cdot 16) \times (N'_{t,y} \cdot 16)$, where the parameters $N'_{t,x}$ and $N'_{t,y}$ are computed via eq. (2) with $S_t = 16$. The block B'_t is then coded using a standard H.264/AVC Intra coder, and the reconstructed pixels are replaced in the region R'_t. Note that pixels in B'_t that do not belong to R'_t are interpolated from the neighboring pixels of R'_t in order to minimize the number of high frequency coefficients.

In case the PSNR obtained approximating R'_t with a plane is higher than T_p, the planar approximation proves to be enough accurate, and therefore, only plane parameters have to be specified. In our implementation, we chose to characterize the plane via 3 points lying on the plane within the block B'_t. Each point is then coded using a Huffman coding table computed from a set of training sequences. The encoder has also to signal to the decoder which sort of coding is applied to the current segment R'_t. This information can be specified using one bit per segment.

2.4. Exploiting the proposed scheme in multi-view plus depth video coding

The proposed scheme can be used to encode a single image with its corresponding depth map but can also be effectively exploited in multi-view plus depth coding schemes. In [8]

a multi-view coding architecture is introduced where all the available views are warped to the viewpoint of one of them using depth information in order to build a stack of aligned views. This stack is then compressed by means of a 3D-DCT transform followed by an entropy coding stage. Regions not visible in the reference view are finally coded using an ad-hoc procedure (refer to [8] for the details). The authors recently proposed a modified version of this approach [11] that uses a single depth map related to the reference view and compresses the stack as a GOP in a video sequence using a standard H.264/AVC encoder instead of the 3D-DCT based approach. As a matter of fact, the depth compression algorithm proposed in this paper can be included into that framework as shown in Fig. 6. We integrated the proposed depth coding scheme into the multi-view encoder in the following way. First, at the encoder side the reference view is compressed using the H.264/AVC Intra coder. Then, the reconstructed view is segmented and the segmentation is employed to code depth data as previously described in this paper. Note that segmentation is computed on the compressed view and can be reproduced at decoder side. As a matter of fact, there is no need to compress and transmit segmentation data. Finally, the compressed depth map is used to warp all the available views to the reference viewpoint and the complete stack can be constructed and encoded using the reference view as the Intra frame and by coding all the other ones as "P" frames.

The decoding can be performed by just reversing the encoding process. First, the stack of views is decompressed and the reference view is extracted. In order to ensure that the extracted reference view is exactly the same that has been segmented at the encoder side, it needs to be encoded in the stack as an Intra picture. For this reason, the views in the stack have been rearranged before the encoding step from the original order $V_1, V_2, \ldots, V_8$ used in [11] to the following one: $V_5, V_4, V_6, V_3, V_7, V_2, V_8, V_1$, with V_5 as the reference view. In this way the reference view is Intra coded and by setting the number of reference frames for motion prediction to 2, it is ensured that all the views in the stack are predicted from at least a neighbor view, allowing the bit rate not to grow. The reference image is coded as an Intra frame and it is the only view that is not affected by the warping process. It can be decompressed without the depth data. Then, the reference view is segmented and the segmentation is used to decompress the depth map using the method presented in this work. Finally, all the other views can be warped to the corresponding viewpoints using the depth map and the original views can be reconstructed.

3. EXPERIMENTAL RESULTS

In order to evaluate the effectiveness of the approach, we compared it with the standard H.264/AVC Intra coder on different depth maps[1]. Since our approach relies on the availability of a coded texture image at the decoder, we adopted DIBR data including both color and depth signals.

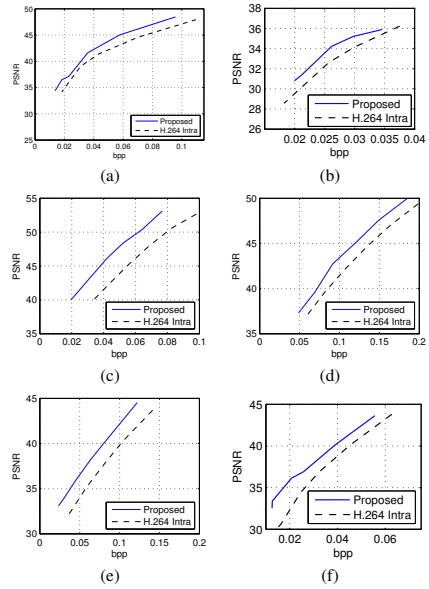


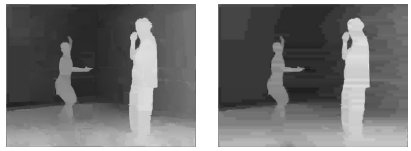
Fig. 7. PSNR vs bpp for frame 0 of depth signals for view 0 of the sequence `breakdancers` (a), view 4 of sequence `ballet` (b), and the stereo images (left view) `baby1` (c) `aloe` (d) `art` (e) `bowling1` (f).

In a first set of experiments, we coded texture images using the standard H.264/AVC Intra coder. The reconstructed views are then segmented by the algorithm in [9]. As a matter of fact, the source coding quality for the color signal affects the performance of the coding algorithm since it changes the set of segments found by the segmentation routine. The quantization parameter for the depth map is found by imposing that the coded bit rate for depth signals is about 20% of the total bit rate[2] (texture and geometry).

Figures 7(a) and 7(b) reports the PSNR vs. bpp for the depth signal of frame 0 of the sequences `breakdancers`

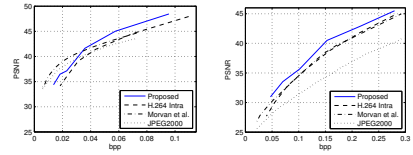
[1]Since the proposed approach is targeted to image coding and there is no exploitation of the inter frame redundancy the results in this section concern only single frame coding. The extension to video sequences will be the subject of future work.

[2]The equalization has been performed coding the depth signal with the H.264/AVC Intra coder.



(a) (b)

Fig. 8. Visual comparison of the compressed depth maps at 0.025 bpp (ballet sequence) (a) proposed scheme; (b) H.264/AVC.



(a) (b)

Fig. 9. PSNR vs bpp curves for depth signals related to frame 0 (view 0) for sequence breakdancers (a) and to left view for image teddy (b). The graphs compare the proposed solution with the H.264/AVC coder, the solution proposed by Morvan *et al.*, and JPEG2000.

and ballet [12]. RD points were found by coding the texture signal with QP varying in the interval $[27, 40]$ and choosing the quantization parameter for the depth signal according to the equalization rule mentioned above. For the sequence breakdancers it is possible to notice that the proposed scheme outperforms the H.264/AVC Intra coder of about 1-2 dB depending on the bit rate. This is mostly due to the fact that the plane approximation characterizes the depth signal more effectively with respect to transform coefficients and spatial prediction. As for the sequence ballet, the proposed scheme improves the quality of the depth map by 1.5 dB on average. A visual comparison of the two methods on this sequence is shown in Fig. 8.

The algorithm was also tested using high quality depth maps included in the stereo sets at [13]. In this set of experiments, we considered the left view of the stereo images baby1, aloe, art, and bowling1 and we generated the R-D plots reported in Fig. 7(c), (d), (e), (f), respectively. In these cases, the captured scene includes a small set of objects which can be easily identified by the segmentation routine even at low bit rates. As a matter of fact, the approximation via planes permits obtaining a higher PSNR values than H.264/AVC. A reduced gain at high bit rates can be noticed for the image ballet in Fig. 7(b). A reason for this can be found in the accuracy of segmentation which may degrade for some regions in the background where edges are not so evident. However, at low bit rates the proposed solution outperforms H.264/AVC since the adoption of planes to approximate depth regions permits coding wide areas at a reasonable quality with a limited bit budget. The approach has also been compared with other coding solutions (see Fig. 9), i.e. JPEG2000 [14] and the platelet-based approach by Morvan *et al.* [2] (using simple piecewise linear functions since it proves to be compatible with our approach where no coefficient prediction is performed). In these tests, we considered the first frame of the sequence breakdancers (view 0) [12] and the image teddy from the Middlebury repository [13]. Experimental results show that the proposed approach outperforms in most situations the compared solutions. On the breakdancers sequence it obtains a slightly better PSNR than H.264/AVC at all bit rates and outperforms the other approaches at bit rates greater than 0.035 bpp. It is possible to notice that al low bit rates the proposed approach is outperformed by the solution by Morvan *et al.* (see Fig. 9a). This fact has to be ascribed to both the quality of original depth map and the source coding quality of the color signal. The original depth information presents several depth estimation errors, while the strong quantization for the color signal introduces several blocking artifacts. As a results, the resulting segmentation poorly fits the depth signal, and therefore, the effectiveness of a segmentation-based coding strategy is slightly compromised. On the teddy image the proposed scheme proves to be the best solution at all the bit rates (see Fig. 9b) obtaining a better PSNR than H.264/AVC and Morvan *et al.* (approximately 1.2 dB higher) while the performance of JPEG2000 on this image is not very impressive. A possible explanation can be found in the fact that JPEG2000 coder was designed for natural images and performs poorly in presence of smooth regions divided by sharp edges. On the contrary, the block-based spatial prediction of H.264/AVC permits characterizing the orientation of depth images along borders more accurately.

Finally, preliminary results on the multi-view coding scheme proposed in Section 2.4 are shown in Fig. 10. In this initial work, we considered only the Luma component. The plot refers to the coding of all the 8 views together with a single depth map and shows how at low bit rates the proposed scheme allows to obtain better results than by coding all the data with H.264 MVC with a maximum gain of about 2 dB.

Moreover, additional tests were performed including temporal prediction for coding entire depth sequences. Preliminary results show that the proposed approach permits improving the coding gain of H.264/AVC since arbitrary segments allow fitting the motion vector field more accurately than standard block matching.

As for the computational complexity of the proposed scheme, the relative coding time increment with respect to H.264/AVC Intra is about 1.35 times higher for the image teddy and 1.22 times higher for the first frame of the se-

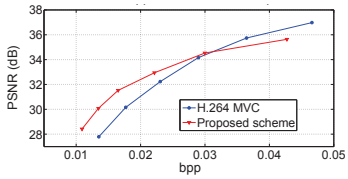


Fig. 10. PSNR vs bpp for frame 0 of DIBR sequence `breakdancers` (8 views plus depth): proposed scheme and H.264 MVC.

quence `breakdancers` (view 4). Note that the code that computes the fitting planes has not been optimized. At the decoder, the complexity increases by 0.12 % with respect to that of the H.264/AVC Intra decoder since the information about segments is not transmitted and has to be derived reproducing the same segmentation routine.

4. CONCLUSIONS AND FUTURE WORK

In this paper a novel coding scheme for depth representations is proposed. The presented algorithm does not code depth as a separate entity from the corresponding texture image but exploits the correlation between color and depth signals to improve compression performances. Segmentation data from the color image permits to predict the shape of the surfaces in the depth maps and to approximate the planar ones with just a few parameters. This information makes possible a significant saving on the coded depth bit rate such that, according to the experimental results, the proposed scheme can outperform H.264 at low bit rates.

Further research will be devoted to the development of a rate-distortion optimization scheme to optimally select the coding strategy for each segment. The inclusion of shape coding techniques will also be investigated in order to improve compression performances on the regions that can not be approximated with a regular surface. Finally, the proposed approach will be extended to the compression of depth video sequences and of multiple depth maps of the same scene.

5. REFERENCES

[1] K.-J. Oh, S. Yea, A. Vetro, and Y.-S. Ho, "Depth reconstruction filter and down/up sampling for depth coding in 3-d video," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 747–750, Sept. 2009.

[2] Y. Morvan, D. Farin, and P. de With, "Depth-Image Compression based on an R-D Optimized Quadtree Decomposition for the Transmission of Multiview Images," in *Proc. of IEEE ICIP 2007*, 2007.

[3] Bo Zhu, Gangyi Jiang, Yun Zhang, Zongju Peng, and Mei Yu, "View synthesis oriented depth map coding algorithm," in *Proc. of APCIP 2009*, july 2009, vol. 2, pp. 104 –107.

[4] Pietro Zanuttigh and Guido M. Cortelazzo, "Compression of depth information for 3D rendering," in *Proceedings of 3DTV 09*, May 2009, pp. 1–4.

[5] S. Milani and G. Calvagno, "A Depth Image Coder Based on Progressive Silhouettes," *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 711–714, Aug. 2010.

[6] M. M. Reid, R. J. Millar, and N. D. Black, "Second-generation image coding: an overview," *ACM Comp. Surv.*, vol. 29, no. 1, pp. 3–29, 1997.

[7] Matthieu Maitre, Yoshihisa Shinagawa, and Minh N. Do, "Wavelet-Based Joint Estimation and Encoding of Depth-Image-Based Representations for Free-Viewpoint Rendering," *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 946–957, 2008.

[8] M. Zamarin, S. Milani, P. Zanuttigh, and G.M. Cortelazzo, "A novel multi-view image coding scheme based on view-warping and 3D-DCT," *Journal Vis. Comm. and Image Representation*, vol. 21, no. 5-6, pp. 462–473, 2010.

[9] P.F. Pedro F. Felzenszwalb and D.P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *IJCV*, vol. 59, no. 2, 2004.

[10] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[11] M. Zamarin, P. Zanuttigh, S. Milani, G. M. Cortelazzo, and S. Forchhammer, "A Joint Multi-View Plus Depth Image Coding Scheme Based on 3D-Warping," in *Proc. of 3DVP 2010*, Oct. 2010, pp. 7–12.

[12] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. of ACM SIGGRAPH 2004*, Los Angeles, CA, USA, Aug. 2004, pp. 600–608.

[13] "Repository vision.middlebury.edu: Stereo datasets," `http://vision.middlebury.edu/stereo`.

[14] David S. Taubman and Michael W. Marcellin, *JPEG2000: image compression fundamentals, standards, and practice*, Kluwer Academic Publishers, Boston, 2002.

A Joint Multi-View Plus Depth Image Coding Scheme Based on 3D-Warping

Marco Zamarin[§], Pietro Zanuttigh[‡], Simone Milani[‡],
Guido M. Cortelazzo[‡], Søren Forchhammer[§]
[§]Technical University of Denmark, Department of Photonics Engineering, Denmark
[‡]University of Padova, Department of Information Engineering, Italy
{mzam,sofo}@fotonik.dtu.dk, {pietro.zanuttigh,simone.milani,corte}@dei.unipd.it

ABSTRACT

Free viewpoint video applications and autostereoscopic displays require the transmission of multiple views of a scene together with depth maps. Current compression and transmission solutions just handle these two data streams as separate entities. However, depth maps contain key information on the scene structure that can be effectively exploited to improve the performance of multi-view coding schemes. In this paper we introduce a novel coding architecture that replaces the inter-view motion prediction operation with a 3D warping approach based on depth information to improve the coding performances. Ad-hoc solutions for occluded areas are also provided. Experimental results show that the proposed joint texture-depth compression approach is able to outperform the state-of-the-art H.264 MVC coding standard performances at low bit rates.

Categories and Subject Descriptors

I.4.1 [**Image Processing and Computer Vision**]: Digitization and Image Capture—*Imaging geometry*; I.4.2 [**Image Processing and Computer Vision**]: Compression (Coding); I.4.10 [**Image Processing and Computer Vision**]: Image Representation—*Multidimensional/Volumetric*

General Terms

Algorithms, Design, Measurements, Performance.

Keywords

3D image coding, multi-view coding, 3D TV, warping, image compression, depth-based rendering

1. INTRODUCTION

Free viewpoint video representation usually requires a large amount of data to represent a dynamic 3D scene. In fact, signals have to characterize both the color (texture) information coming from multiple camera streams and geometry (depth) data, which are needed to perform the 3D warping operation. This representation, referred in literature as "multi-view plus depth", permits creating different views in addition to the already-available ones as if a standard 3D representation of the scene was available. However, the versatility and the flexibility of the browsing experience strictly depend on the number of views and depth maps, and their compression is a fundamental step in the construction of a practical system.

The color information is usually the largest part of the data (e.g. Ozaktas and Onural in [9] suggest to allocate 80% of the total bit rate to color information). Consequently, different solutions were proposed to efficiently compress such data also exploiting the redundancy between different views of the same scene. Probably the most popular solution is H.264 MVC (Multiview Video Coding) [3], which has been derived from the standard video coder H.264 and permits compressing multi-view representations in an effective way. Other approaches are based on multi-dimensional wavelet transforms, like the method of [13] which combines a 4D wavelet transform with an *ad-hoc* disparity-compensated view filter (DCVF). Another possibility is to exploit 3D warping of the temporal-motion vectors in one view in order to predict the motion vectors in the other ones like in [2]. Depth information can be compressed by standard image or video coding tools like H.264 or by *ad-hoc* methods that exploit the peculiar characteristics of this kind of data. These solutions can rely on a careful handling of edge-regions [7], [14] or on the conversion of the depth map to a 3D mesh [1]. However, excluding few recent papers like [4] (where a wavelet based scheme is proposed to jointly encode depth and texture), most of the proposed schemes for multi-view plus depth representation handle the compression of the two data streams as separate tasks. The geometrical description of the scene given by the depth data can instead be used to understand the relationship between the various views and to replace prediction schemes based on motion vectors (see [3]).

This work introduces a novel compression scheme for multi-view plus depth representation based on 3D warping. Depth information is used to warp the various views to build a stack of aligned images that can then be efficiently compressed. *Ad-hoc* strategies are proposed to fill the occluded regions.

The rest of the paper is organized in the following way: Section 2 presents the proposed coding algorithm, Section

3DVP'10, October 29, 2010, Firenze, Italy.

M. Zamarin, S. Milani, P. Zanuttigh, G.M. Cortelazzo, S. Forchhammer, "A Joint Multi-View Plus Depth Image Coding Scheme Based on 3D-Warping", *Proc. of the 1st ACM Int'l Work. on 3D Video Proc. (3DVP 2010)*, pp. 7-12, Florence, Italy, Oct. 29, 2010.

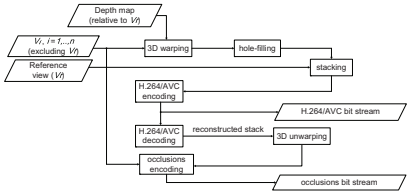


Figure 1: Encoder block scheme

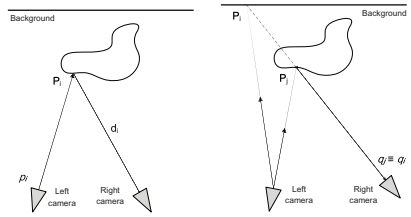


Figure 2: Warping procedure: a) Warping of a single sample; b) Occlusion detection

3 discusses the experimental results, and finally, Section 4 draws the final conclusions.

2. PROPOSED CODING ALGORITHM

The proposed coding scheme aims at encoding stereo or multi-view images exploiting the geometric description of the scene. More precisely, the geometric information is provided by a depth map referring to the viewpoint of one of the views (denoted as the *reference* view V_r in the rest of the paper). We will show how depth data permit exploiting the inter-view redundancy more effectively with respect to the block-based prediction of current multi-view video coding schemes.

The proposed algorithm encompasses three main steps. In the first step a reference viewpoint is chosen and all the available views are warped with respect to it in order to build a stack of aligned camera pictures. This task can be performed using only one depth map related to the reference viewpoint and permits obtaining a very compact representation since in the warped views all the pixels corresponding to the same 3D point are placed at the same location. The stack is then further processed and encoded exploiting the H.264/AVC coder. In the last step, the occluded regions (the ones visible in some side views but not in the reference one) are handled with an *ad-hoc* coding scheme.

The block diagram of the proposed coding scheme is shown in Fig. 1.

2.1 3D-warping operations

The first step of the proposed encoding scheme consists in creating a stack of images to be encoded by means of 3D-warping. The stack is built by reprojecting all the available views over the reference one. We will assume to have the depth map corresponding to the reference view while no depth information is available for the other ones. For example in the stereo (2 views) case, only one depth map (e.g., the one referring to the left view) is available. This configuration can be simply extended to the multi-view case, with more views and a still a single depth map related to one viewpoint (the reference view). The warping procedure will be described with respect to the stereo case, but it can be easily extended to the multi-view case by performing the same operations for all the views except for the reference one. The warping is performed in two different ways in order to deal more effectively with the occluded regions and to account for the fact that only the depth data corresponding to the reference viewpoint is available. More precisely, *backward mapping* is employed in the warping stage while *forward mapping* is used for the unwarping.

Let us consider at first the warping process. For each pixel p_i in the left (reference) view, the corresponding 3D-point $\mathbf{P}_i$ is computed using the relative depth information. A warping procedure is then performed in order to project the right view to the viewpoint of the left one according to the following steps:

1. All the distance values d_i between the right camera and the 3D-points are sorted.
2. Starting from the smallest distance, each pixel p_i of the reference view is warped to the right view obtaining the corresponding location q_i'.
3. The coordinates of the obtained pixel q_i' are rounded leading to an actual pixel q_i for the right view.
4. If location q_i has already been selected to fill a pixel of the warped image, i.e. $q_i \equiv q_j$ and $p_i \not\equiv p_j$, the previous value is kept. Since the warping is performed starting from the objects closer to the right camera, if p_i is mapped to an already selected location q_i it implies that the 3D point $\mathbf{P}_i$ lies along the optical ray joining the right camera and the 3D-point $\mathbf{P}_j$ but is placed *behind* $\mathbf{P}_j$ (Fig. 2b). As a consequence, the right camera can not see it.
5. The color corresponding to the location q_i' is approximated using a bilinear interpolation of the four closest samples in the right view (the warped location q_i' has non-integer coordinates) and copied into the warped image at the location of p_i.

Note that this procedure requires only the depth data associated to the reference viewpoint. The inverse warping (unwarping) is made in a kind of opposite way using *forward mapping* (*backward mapping* can not be used because depth data is available only for the reference view). In this case the problem can be simply solved by warping the samples ordered according to decreasing distances. Even if in the unwarped images some pixels are filled more than once (because of the unwarping of occluded regions), the processing of pixels from the furthest to the closest distances with respect to the right camera ensures that the objects in the foreground are correctly rendered. These is due to the fact that closer points are rendered at the end and the farther ones are overwritten). Because of rounding operations some artifacts may appear. In this case, a hole-filling step based

a)

b)

Figure 3: View warping: a) Right view (V_6) warped with respect to the left one (V_7) and b) relative filled version. Note that large holes in the first image are due to occluded regions while the smaller ones on the floor are due to the artifacts on the source depth map produced by the 3D reconstruction algorithm. All of the missing pixels have been filled with the method described in Section 2.2.

on median filtering is applied to small holes. Note also that some samples (the ones not visible from the reference view) can not be filled and need to be reconstructed following a different strategy (see Section 2.3). As an example, Fig. 3 concerns a stereo setup made by views 6 and 7 of the sequence `breakdancers` provided by Microsoft [15] and shows view 7 warped over view 6 before and after the filling process.

2.2 Data encoding

The warping process described in the previous section is applied to all the available views excluding the reference one in order to build a stack of aligned views (as shown in Fig. 4). The images related to the different viewpoints are warped generating a set of aligned views, where each pixel in the reference picture is associated to a pixel in the other side images. This stack can be compressed very efficiently due to the alignment at the same location of pixels associated to the same 3D point. As expected, the accuracy of this association strongly depends on the accuracy of the depth map information (which can be distorted by both source coding operations and errors in the 3D reconstruction). This mismatch has to be compensated by a residual coding unit,

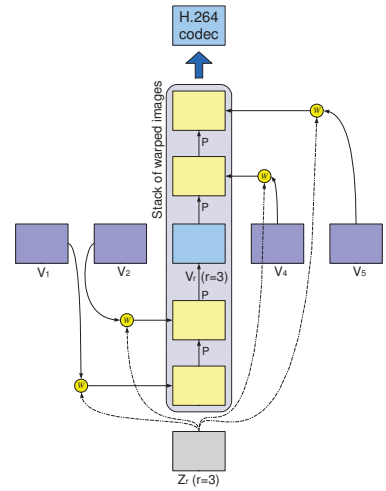


Figure 4: Stack creation and encoding for a multi-view image with 5 views. Labels "*W*" in the scheme indicate a warping operation involving a single view and the depth map relative to the reference view.

which characterizes the prediction error between the reference view and its warped version obtained from the signal to be coded, and by an occlusion coding unit, which compresses those parts of the video signal that are required to complete the reconstructed image. In the following, we will describe the residual coding unit, while the whole procedure that handles and codes the occluded regions will be presented in the following section.

The warping operation generates a stack of k images corresponding to the warped versions of the original $k-1$ side views to be coded and the reference view itself. These images are then coded using the H.264/AVC coding engine considering the k different images as temporally-adjacent frames in a video sequence. These pictures present several holes including those pixels p_i of the reference view that can not be associated to a pixel q_i in the side views. As a matter of fact, a pre-processing stage is needed in order to fill the remaining holes. The presence of discontinuities in the coded signal produces many high-frequency coefficients in the transform domain, which alter the low-pass characteristics of traditional image signals and increase the required bit rate. This inconvenience can be avoided by linear pixel interpolation along the "view" dimension of the stack. Given a stack of k input views, the coding algorithm creates a k-dimensional array $\mathbf{b}(x, y, .)$ by grouping the pixels in the warped views at the coordinates (x, y) (following the positioning order of the cameras). Whenever a missing pixel is revealed, it is filled by linear interpolation of the two adjacent available pixels in the array $\mathbf{b}(x, y, .)$. Note that at least one pixel is available in $\mathbf{b}(x, y, .)$, since the reference view does not have missing

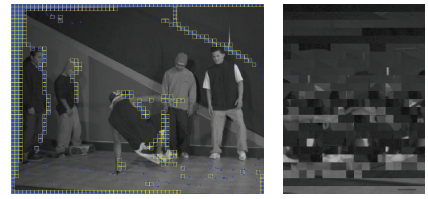
pixels. In case only one neighbor is available (e.g. for stereo sequences where $k = 2$), the pixel value is copied from the closest available view. Figure 3b) shows an example of this situation.

After the warping operation, the sequence of warped views is coded using a standard H.264/AVC codec [12]. The current warped view is partitioned into 8×8 blocks $\mathbf{x}$, and the motion estimation (ME) unit approximates each block $\mathbf{x}$ looking for a predictor block $\mathbf{x}_p$ in the previously-coded views that minimizes a Lagrangian distortion metric. The predictor block $\mathbf{x}_p$ can be identified by a motion vector, which could present non-zero coordinates in case the ME unit has to compensate eventual inaccuracies in the warping operations due to the distortion in the coded depth maps and to the limited accuracy of the 3D reconstruction. However, since after the 3D warping most of motion vectors identify a null displacement, the required bit rate to code motion information is significantly smaller. The prediction residual block $\mathbf{e} = \mathbf{x} - \mathbf{x}_p$ is then transformed into the block $\mathbf{E}$ and quantized into the transform coefficients $\mathbf{E}_q$. The block $\mathbf{E}_q$ is then converted into a binary bit stream, together with motion vectors, coded block pattern (CBP) information, and coding modes for the current block. Transform coefficients are then dequantized and inversely-transformed into a distorted residual signal $\mathbf{e}_r$, which permits reconstructing the coded block $\mathbf{x}_r$ of the warped image via the addition $\mathbf{x}_r = \mathbf{e}_r + \mathbf{x}_p$. In short, these coding operations correspond to a standard H.264/AVC Inter coding. The original view can then be reconstructed by an inverse warping operation, and the occluded regions can be handled as described in the next section.

2.3 Handling of occluded regions

The coding algorithm presented in the previous section permits coding effectively only the samples that correspond to a pixel in the reference view via a warping operation. After decoding, there are missing pixels in the other views that need to be represented as well.

Connected regions of missing pixels are referenced as "holes", and they can be interpolated from the available neighboring pixels (whenever convenient) or coded independently. Both solutions are not completely satisfactory. Interpolating the missing samples can lead to a poor image quality as the holes grow larger. On the other hand, an independent encoding of all the missing samples increases the coded bit rate and proves to be inefficient for small holes, which can be easily interpolated introducing a limited distortion. From these premises, we adopted a hybrid approach that enacts one or the other coding strategy for missing pixels in the unwarped images according to the size of holes. At first, the coding routine counts the number of pixels in each hole. The pixel number is compared with a threshold t_r, which has been found experimentally. Figure 5a shows an example of unwarped image with the missing samples highlighted in blue, where it can be noticed the presence of both isolated holes and larger regions. The latter can be both occluded areas and side regions that are out of the reference camera field of view. The regions smaller than t_r are simply interpolated on the basis of the surrounding available pixels. The other ones are instead coded separately. A set of 16×16 macroblocks is selected from the original side views by taking the macroblock regions covering the large holes to be coded (the macroblock regions are highlighted in yellow



a) b)

Figure 5: Hole filling scheme: a) Macroblocks corresponding to large holes (the view is the unwarped version of the image in Fig. 3b) and b) relative "Macroblock image" (scaled). Blue pixels refer to occluded regions. Note that small holes are not filled using macroblocks.

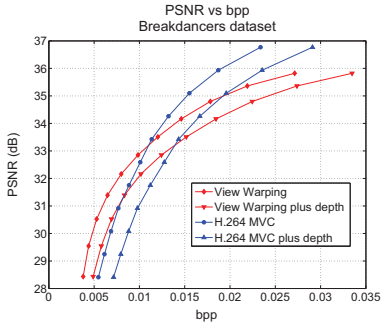
in Figure 5a). The selected macroblocks are then packed inside a still image (see Fig. 5b) that is finally coded using the standard H.264/AVC coder at the same quality level of the already-coded data. In our experiments, we found that a threshold $t_r = 36$ provides a good trade-off between the bit rate and the image quality. However, the value of t_r can be varied depending on the required rate vs. quality trade-off.

3. EXPERIMENTAL RESULTS

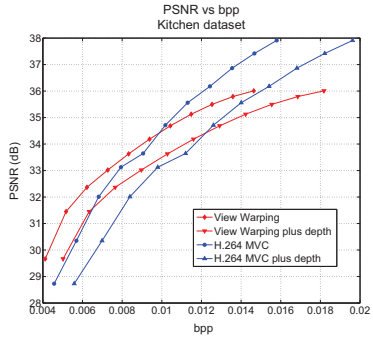
In order to evaluate the performance of the proposed solution, we tested the presented coding scheme with different configurations and different test images. More precisely, we considered the sequence `breakdancers` from Microsoft Research [15] and the set `kitchen` generated from a 3D synthetic model[1]. Each set is made of 8 different views (resolution 1024×768) taken by an array of cameras placed on an arc-shaped alignment around the scene (for the `breakdancers` sequence only the first frame was considered). In the coding process, the central view was adopted as reference view, and only the relative depth map is employed in the warping process. As a matter of fact, while the number of views can vary according to the desired configuration, the employed depth map is always associated to a single view.

In our tests, at first we considered a Free Viewpoint Video (FVV) setting where all 8 views are transmitted, together with the depth map of view 4. In a second configuration, we tested the performance of the scheme for a stereo video coding scheme involving a two-views-plus-depth signal. Depth maps were independently coded using a standard H.264/AVC Intra-only codec. As a matter of fact, the bit rate required to code the depth information has a significant impact on the final bandwidth usage. Moreover, the quality of the depth map is related to the quality of the coded views. Each set (both views and depth maps) was coded using a fixed quantization parameter QP. For depth map signals, the adopted QP is selected in order to allocate about one fourth of the available bandwidth to depth map coding (as suggested in [11]). Figure 6 reports the PSNR (dB) vs. bit-per-pixel (bpp) for the joint compression of all the 8 views of the sequences `breakdancers` and `kitchen` (multi-view settings).

[1]The full `kitchen` dataset is available at `http://lttm.dei.unipd.it/downloads/kitchen`



a)



b)

Figure 6: Coding performance comparisons on a) `breakdancers` 8-view dataset and b) `kitchen` 8-view dataset

Labels `View Warping` and `H.264 MVC` refers to the results for the proposed approach and for H.264 MVC respectively, while labels with the suffix "`plus depth`" refers to plots that include the bit cost for depth information (i.e. a single depth map corresponding to V_r).

Points are calculated considering the average PSNR within all the 8 reconstructed views in each sequence.

As for the `breakdancers` sequence (Fig. 6a) at low bit rates the PSNR of the reconstructed sequence by the proposed approach improves with respect to the H.264/AVC (our method achieves a 1.2 dB gain over H.264/AVC at 0.0075 bpp) since the higher quantization steps combined with the warping-based approach minimize the amount of information that has to be coded after warping (compare the curve labelled `View Warping` with the curve `H.264 MVC` in Fig. 6a). A visual comparison between our method and H.264/AVC is provided in Fig. 7. On the contrary, at high bit rates the H.264 MVC coder outperforms the proposed scheme since the warping operation does not permit an enough accurate approximation of the reference view

a) b)

Figure 7: Visual comparison (`breakdancers` 8-view dataset, view 4 at 0.01 bpp) : a) H.264 encoding and b) proposed method

and the coding of the occluded regions requires a significant amount of bits. Considering only the bit rates of coded texture data, it is possible to notice that the performance of the proposed solution becomes competitive at 0.011 bpp (if we consider the curves including also the depth bit rate the crossing is at 0.0135 bpp at the same 33.2 dB PSNR). If a depth map needs to be transmitted just for the compression of a small number of views, the proposed approach is less appealing but can still be competitive at very low bit rates (compare the curve labelled `View Warping plus depth` with the curve `H.264 MVC` in Fig. 6a). As for the `kitchen` sequence (Fig. 6b), it is possible to notice that the warping operation is more effective since the depth maps are generated from a synthetic model. As a consequence, the rate-distortion performance of the warping-based approach becomes competitive at a higher average quality (corresponding to a 34.4 dB PSNR), but the distance between the H.264 MVC and the warping-based curves is smaller compared to the `breakdancers` sequence since the positioning of cameras[2] generates a greater amount of occluded regions that have to be filled with Intra coded pixels (see Section 2.3).

Since the original views have different distances from the reference one, the reconstructed quality differs among the views decreasing with increasing distance. As an example, at an average PSNR of 31.4 dB, the standard deviation of the PSNR among all the views is 1.28 dB for the `breakdancers` sequence.

As for the stereo setting, experimental results for the `breakdancers` sequence are reported in Fig. 8. In this configuration the proposed approach proves again to be competitive at low bit rates (less than 0.015 bpp), despite the PSNR gains over the traditional H.264 MVC approach are smaller with respect to the multi-view case. For the stereo configuration we have a maximum gain of 1 dB at 0.0075 bpp. Note that the transmitted depth map has a stronger impact on the overall bit rates since in the multi-view case the weight of the coded bits for one depth map is distributed on 8 views while in the stereo case there is still a single depth map but with just two views. As a matter of fact, the transmission of the depth information just for color encoding purposes is not rewarding in the stereo case whenever the viewing display or application does not require it (compare the curve labelled `View Warping plus Depth` with the curve `H.264 MVC` in Fig. 8). In case depth information is needed, experimental results show that the employment of geometric information in encoding color data significantly improves the coding per-

[2]In the `kitchen` sequence the cameras are farther one from the other than in the `breakdancers` one

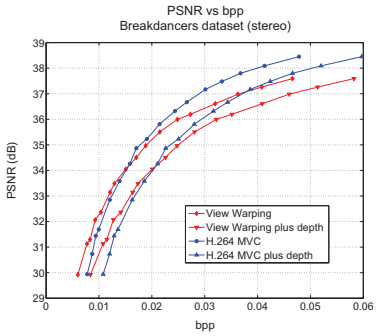


Figure 8: Coding performance comparisons on the `breakdancers` 2-view dataset

formance (compare the curve labelled `View Warping plus Depth` with the curve `H.264 MVC plus depth`).

4. CONCLUSION AND FUTURE WORK

In this paper a novel coding scheme for multi-view plus depth representations is proposed. Depth information is used to reproject all the available views to a common viewpoint in order to build a stack of aligned views that can be compressed very efficiently using standard video coders like H.264. Novel interpolation and filling schemes, which handle the samples visible from one viewpoint only, are also proposed and used in both the stack creation and in the reconstruction of the source views. Experimental results proved how the proposed scheme can outperform H.264 MVC at very low bit rates even if an additional depth map has to be transmitted to aid the coding of the texture data. In free viewpoint video applications, where depth data need to be transmitted in any case permitting an interactive browsing of the scene, the proposed scheme becomes even more appealing since it obtains a relevant gain on the transmission of the depth and texture using H.264 MVC at low bit rates.

Further research will be devoted to the inclusion of *ad-hoc* compression schemes for depth information into the proposed framework in order to reduce the coded bit stream. We will consider different algorithms, including the possibility of using block-based search routines and of compressing the resulting map through H.264/AVC, like in [5].

The warping process can also be improved adopting a finer filling procedures for occluded regions as shown in [6]. A more careful handling of occluded regions should reduce the overall bit rate. To this purpose, an interesting possibility is to manipulate each occlusion in such a way that the neighboring pixels only belong to the background region (as done in [8]). Concerning the encoding of the image stack, an approach based on shape coding will be investigated. The joint exploitation of the temporal and inter-view redundancy, together with a joint rate-distortion optimization, will also be addressed in order to find an optimal balance of the data allocation on both texture and depth bit streams. The rate-distortion optimization method proposed in [10] could be selected as a good starting point. Even if the current approach does not include particularly time-demanding steps, efficiency and complexity issues will also be taken into account for the future development.

5. REFERENCES

[1] B.-B. Chai, S. Sethuraman, H. S. Sawhney, and P. Hatrack. Depth map compression for real-time view-based rendering. *Pattern Recogn. Lett.*, 25(7):755–766, 2004.

[2] X. Guo, Y. Lu, F. Wu, and W. Gao. Inter-View Direct Mode for Multiview Video Coding. *IEEE Trans. Circuits Syst.*, 16(12):1527–1532, Dec. 2006.

[3] ISO/IEC MPEG & ITU-T VCEG. Joint Draft 8.0 on Multiview Video Coding, Jul. 2008.

[4] M. Maitre, Y. Shinagawa, and M. N. Do. Wavelet-Based Joint Estimation and Encoding of Depth-Image-Based Representations for Free-Viewpoint Rendering. *IEEE Trans. Image Process.*, 17(6):946–957, 2008.

[5] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun. Extensions of H.264/AVC for Multiview Video Compression. In *Proc. of IEEE ICIP 2006*, pages 2981–2984, Oct. 2006.

[6] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto. View generation with 3D warping using depth information for FTV. *Image Commun.*, 24(1-2):65–72, 2009.

[7] Y. Morvan, D. Farin, and P. de With. Depth-Image Compression based on an R-D Optimized Quadtree Decomposition for the Transmission of Multiview Images. In *Proc. of IEEE ICIP 2007*, 2007.

[8] K.-J. Oh, S. Yea, and Y.-S. Ho. Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video. In *Proc. of PCS 2009*, pages 233–236, Piscataway, NJ, USA, 2009.

[9] H. Ozaktas and L. Onural. *Three-Dimensional Television: Capture, Transmission, Display*. Springer, 2008.

[10] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima. View Scalable Multiview Video Coding Using 3-D Warping With Depth Map. *IEEE Trans. Circuits Syst. Video Technol.*, 17(11):1485 –1495, nov. 2007.

[11] A. Smolic, H. Kimata, and A. Vetro. Development of MPEG standards for 3D and free viewpoint video. In *Three-Dimensional TV, Video, and Display IV, Proceedings of SPIE*, volume 6016, Oct. 2005.

[12] T. Wiegand. Version 3 of H.264/AVC. In *Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, 12^{th} Meeting*, Redmond, WA, USA, July 17 – 23, 2004.

[13] W. Yang, Y. Lu, F. Wu, J. Cai, K. N. Ngan, and S. Li. 4-D Wavelet-Based Multiview Video Coding. *IEEE Trans. Circuits Syst. Video Technol.*, 16(11):1385–1396, Nov. 2006.

[14] P. Zanuttigh and G. M. Cortelazzo. Compression of depth information for 3D rendering. In *Proceedings of 3DTV 09*, pages 1–4, May 2009.

[15] L. C. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608, 2004.

Appendix B

Test Material

The first frame of each test sequence used for the experiments is shown below.

B.1 Multi-View Video-plus-Depth Sequences

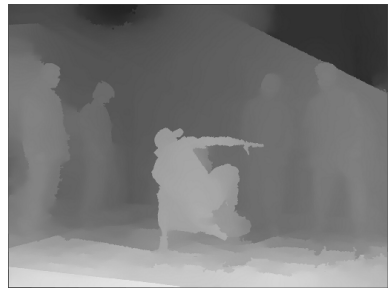
Texture

Depth

Figure B.1: *Balloons*, view 3, 1024 × 768, 30 fps.


Texture Depth

Figure B.2: *BookArrival*, view 8, 1024 × 768, 16.67 fps.

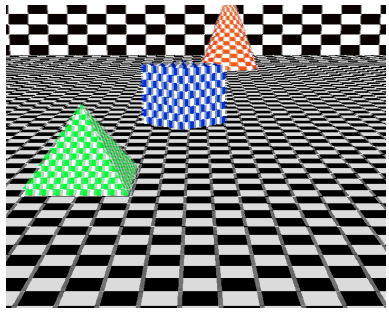



Texture Depth

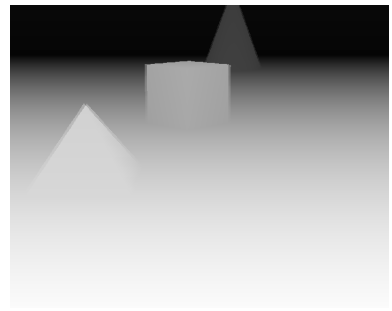
Figure B.3: *Breakdancers*, view 4, 1024 × 768, 15 fps.


Texture Depth

Figure B.4: *Cafe*, view 3, 1920 × 1080, 30 fps.



Texture



Depth

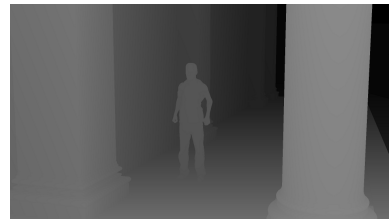
Figure B.5: *Cg*, 720 × 576, 25 fps, interlaced.

Texture

Depth

Figure B.6: *ChampagneTower*, view 37, 1280 × 960, 29.41 fps.

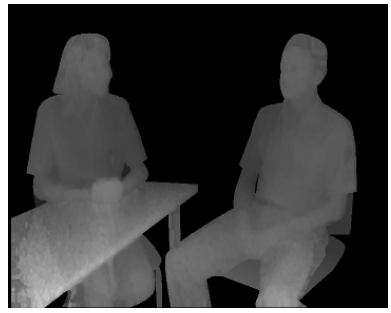
Texture



Depth

Figure B.7: *Dancer*, view 5, 1920 × 1088, 25 fps.

Texture



Depth

Figure B.8: *Interview*, view 5, 720 × 576, 25 fps, interlaced.

Texture

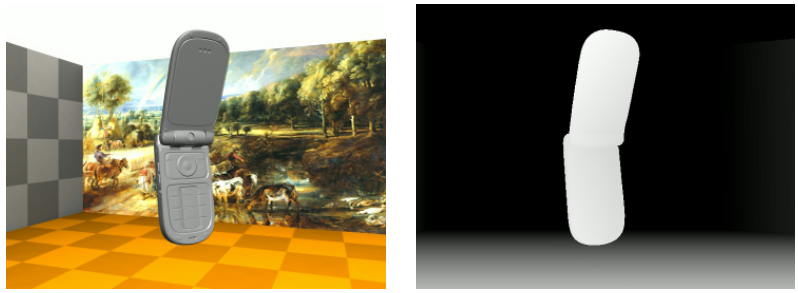
Depth

Figure B.9: *Kendo*, view 3, 1024 × 768, 30 fps.

Texture

Depth

Figure B.10: *Lovebird1*, 1024 × 768, 30 fps.



Texture Depth

Figure B.11: *Mobile*, view 4, 720 × 540, 30 fps.

Texture Depth

Figure B.12: *Newspaper*, view 4, 1024 × 768, 30 fps.

Texture Depth

Figure B.13: *Orbi*, 720 × 576, 25 fps, interlaced.

B.2 Multi-View Images plus Depth

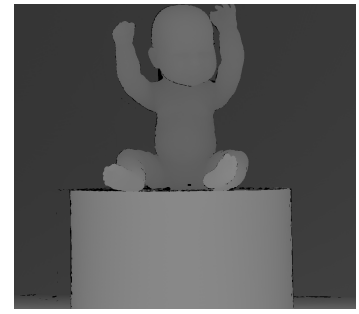
Texture Depth

Figure B.14: *Aloe*, view 1, 1282 × 1110.

Texture Depth

Figure B.15: *Art*, view 1, 1390 × 1110.

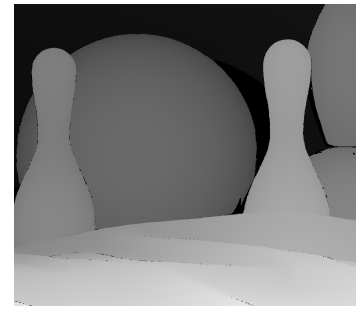
Texture



Depth

Figure B.16: *Baby1*, view 1, 1240 × 1110.

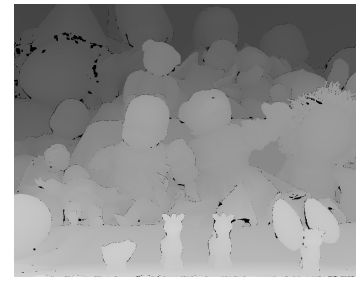
Texture



Depth

Figure B.17: *Bowling1*, view 1, 1252 × 1110.

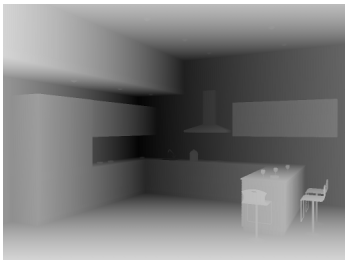
Texture



Depth

Figure B.18: *Dolls*, view 1, 1390 × 1110.

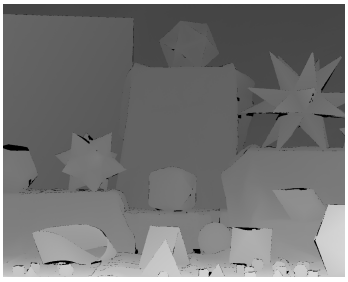
Texture



Depth

Figure B.19: *Kitchen*, view 4, 1024 × 768.

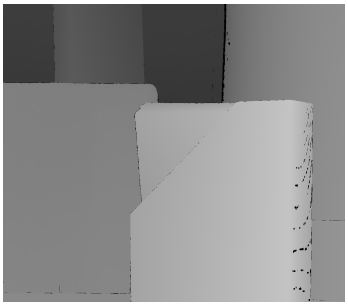
Texture



Depth

Figure B.20: *Moebius*, view 1, 1390 × 1110.

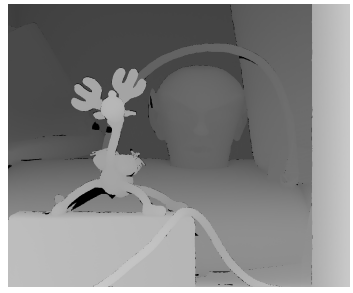
Texture



Depth

Figure B.21: *Plastic*, view 1, 1270 × 1110.

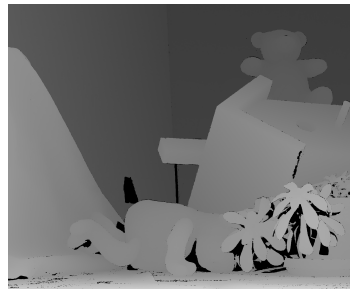
Texture



Depth

Figure B.22: *Reindeer*, view 1, 1342 × 1110.

Texture



Depth

Figure B.23: *Teddy*, view 2, 1800 × 1500.

List of Acronyms

3DTV	Three-Dimensional Television
3DV	Three-Dimensional Video
3D-DCT	3D Discrete Cosine Transform
ARPS	Adaptive Rood Pattern Search
AVC	Advanced Video Coding
CABAC	Context-based Adaptive Binary Arithmetic Coding
DCT	Discrete Cosine Transform
DERS	Depth Estimation Reference Software
DIBR	Depth-Image-Based-Rendering
DVB-H	Digital Video Broadcasting – Handheld
DVC	Distributed Video Coding
FDIS	Final Draft International Standard
FVV	Free Viewpoint Video
GOP	Group Of Pictures
HEVC	High Efficiency Video Coding
HD	High Definition
IEC	International Electrotechnical Commission

ISO	International Organization for Standardization
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
JCT-3V	Joint Collaborative Team on 3D Video Coding Extension Development
JTC	Joint Technical Committee
LDI	Layered Depth Image
MDC	Multiple Description Coding
MPEG	Moving Pictures Experts Group
MSE	Mean Squared Error
MV	Motion Vector
MVC	Multi-view Video Coding
MVD	Multi-view Video-plus-Depth
OBMC	Overlapped Block Motion Compensation
PDE	Partial Differential Equation
PSNR	Peak Signal-to-Noise Ratio
QP	Quantization Parameter
RD	Rate-Distortion
SC	Sub-Committee
SG	Study Group
SI	Side Information
ToF	Time-of-Flight
VSRS	View Synthesis Reference Software
WG	Working Group
WP	Working Party

Bibliography

[1] C. Wheatstone, "Contributions to the physiology of vision - Part the first. On some remarkable, and hitherto unobserved, phenomena of binocular vision," *Philosophical Transactions of the Royal Society of London*, vol. 128, pp. 371–394, 1838.

[2] C. Fehn, "3D TV Broadcasting," in *3D Videocommunication*, O. Schreer, P. Kauff, and T. Sikora, Eds. Wiley & Sons Ltd, 2005, ch. 2, pp. 23–38.

[3] ITU-T and ISO/IEC JTC1, Final draft amendment 3, Amendment 3 to ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ISO/IEC JTC 1/SC 29/WG 11 (MPEG) Doc. N1366, Sep. 1996.

[4] ITU-T and ISO/IEC JTC 1, Generic coding of moving pictures and associated audio information - Part 2: Video, ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video), 1994.

[5] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[6] A. Vetro, P. Pandit, H. Kimata, A. Smolic, and Y.-K. Wang, Joint Draft 8 of Multiview Video Coding, Hannover, Germany, Joint Video Team (JVT) Doc. JVT-AB204, Jul. 2008.

[7] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.

[8] C. Fehn, "Depth-Image-Based Rendering (DIBR), Compression and Transmission for a New Approach on 3D-TV," in *2004 SPIE Stereosc. Display Virt. Reality Syst. XI*, 2004, pp. 93–104.

[9] Video and Requirements Group, Vision on 3D Video, ISO/IEC JTC1/SC29/WG11 N10357, Lausanne (CH), Feb. 2008.

[10] ISO/IEC JTC 1/SC 29/WG 11, Press Release of the 103rd Meeting in Geneva, Switzerland, Doc. N13253, Geneva (CH), Jan. 2013.

[11] K. Müller, P. Merkle, and T. Wiegand, "3-D Video Representation Using Depth Maps," *Proc. IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[12] P. Kauff, N. Atzpadin, C. Fehn, M. Müller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3DTV services providing interoperability and scalability," *Signal Process.: Image Commun.*, vol. 22, no. 2, pp. 217–234, 2007.

[13] L. Onural, "State-of-the-Art in 3D Imaging, Delivery, and 3D Content Display," in *3D Video Technologies*. SPIE Press, 2011, ch. 3, pp. 23–60.

[14] ISO/IEC JTC1/SC29/WG11, Applications and Requirements on 3D Video Coding, Doc. N12035, Geneva (CH), Mar. 2011.

[15] A. Bogomjakov, C. Gotsmann, and M. Magnor, "Free-Viewpoint Video from Depth Cameras," in *2006 Vision, Modeling and Visualization (VMV 2006)*, Nov. 2006, pp. 89–96.

[16] M. Tanimoto, "Free Viewpoint Systems," in *Three-Dimensional Television*. Wiley & Sons, Ltd, 2005, ch. 4, pp. 55–73.

[17] L. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, 2004.

[18] G. Cheung, A. Ortega, Q.-S. Kim, V. Velisavljevic, and A. Kubota, "Depth Map Compression for Depth-Image-Based Rendering," in *3D-TV System with Depth-Image-Based Rendering*, C. Zhu, Y. Zhao, L. Yu, and M. Tanimoto, Eds. Springer, 2013, ch. 9, pp. 249–276.

[19] C. Zhu, Y. Zhao, L. Yu, and M. Tanimoto, Eds., *3D-TV System with Depth-Image-Based Rendering*. Springer Heidelberg, 2013.

[20] H. M. Ozaktas and L. Onural, Eds., *Three-Dimensional Television*. Springer Heidelberg, 2008.

[21] O. Schreer, P. Kauff, and T. Sikora, Eds., *3D Videocommunication*. Wiley & Sons, Ltd, 2005.

[22] L. Onural, *3D Video Technologies, An Overview of Research Trends*. SPIE Press, 2011.

[23] S. Kang, R. Szeliski, and P. Anandan, "The geometry-image representation tradeoff for rendering," in *2000 IEEE Int'l Conf. Image Process. (ICIP 2000)*, vol. 2, Sep. 2000, pp. 13–16.

[24] H.-Y. Shum, S. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1020–1037, Nov. 2003.

[25] A. Smolic, P. Merkle, K. Müller, C. Fehn, P. Kauff, and T. Wiegand, "Compression of Multi-view Video and Associated Data," in *Three-Dimensional Television*. Springer Heidelberg, 2008, ch. 9, pp. 313–350.

[26] R. Koch and J.-F. Evers-Senne, "View Synthesis and Rendering Methods," in *3D Videocommunication*. Wiley & Sons, Ltd, 2005, ch. 4, pp. 55–73.

[27] E. H. Adelson and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision," in *Computational Models of Visual Processing*. MIT Press, 1991, pp. 3–20.

[28] J. Shade, S. Gortler, L. He, and R. Szeliski, "Layered Depth Images," in *Conf. Comput. Graphics Interactive Techn.*, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 231–242.

[29] A. Frick, B. Bartczak, and R. Koch, "Real-time preview for layered depth video in 3D-TV," in *2010 SPIE Real-Time Image Video Process.*, May 2010, pp. 77 240F/1–10.

[30] S. Ivekovic, A. Fusiello, and E. Trucco, "Fundamentals of Multiple-view Geometry," in *3D Videocommunication*. Wiley & Sons, Ltd, 2005, ch. 6, pp. 93–113.

[31] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[32] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *1997 IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR 1997)*, Jun. 1997, pp. 1106–1112.

[33] ISO/IEC JTC1/SC29/WG11, Text of ISO/IEC FDIS 23002-3 Representation of Auxiliary Video and Supplemental Information, Doc. N8768, Marrakech, Morocco, Jan. 2007.

[34] ISO/IEC JTC1/SC29/WG11, Text of ISO/IEC 13818-1:2003/FDAM2 Carriage of Auxiliary Data, Doc. N8799, Marrakech, Morocco, Jan. 2007.

[35] M. Tanimoto, T. Fujii, K. Suzuki, N. Fukushima, Y. Mori, Reference softwares for depth estimation and view synthesis, ISO/IEC JTC 1/SC 29/WG 11 Doc. M15377, Archamps, France, Apr. 2008.

[36] M. Gotfryd, K. Wegner, M. Domanski, View Synthesis Software and assessment of its performance, ISO/IEC JTC 1/SC 29/WG 11 Doc. N15672, Hannover, Germany, Jul. 2008.

[37] P. Rana and M. Flierl, "Depth consistency testing for improved view interpolation," in *2010 IEEE Int'l Work. Multimedia Signal Process. (MMSP 2010)*, Oct. 2010, pp. 384–389.

[38] W. Sun, O. Au, L. Xu, S. Chui, C. Kwok, and Y. Li, "Error compensation and reliability based view synthesis," in *2011 IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP 2011)*, May 2011, pp. 1349–1352.

[39] L. Yang, M. Wildeboer, T. Yendo, M. Tehrani, T. Fujii, and M. Tanimoto, "Reducing bitrates of compressed video with enhanced view synthesis for FTV," in *2010 Picture Coding Symp. (PCS 2010)*, Dec. 2010, pp. 5–8.

[40] H. Yuan, J. Liu, H. Xu, Z. Li, and W. Liu, "Coding Distortion Elimination of Virtual View Synthesis for 3D Video System: Theoretical Analyses and Implementation," *IEEE Trans. Broadcast.*, vol. 58, no. 4, pp. 558–568, Dec. 2012.

[41] I. Daribo and H. Saito, "Bilateral depth-discontinuity filter for novel view synthesis," in *2010 IEEE Int'l Work. Multimedia Signal Process. (MMSP 2010)*, Oct. 2010, pp. 145–149.

[42] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video," in *2009 Picture Coding Symp. (PCS 2009)*, May 2009, pp. 1–4.

[43] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *2010 IEEE Int'l Work. Multimedia Signal Process. (MMSP 2010)*, Oct. 2010, pp. 167–170.

[44] H. Lim, Y. Kim, S. Lee, O. Choi, J. Kim, and C. Kim, "Bi-layer inpainting for novel view synthesis," in *2011 IEEE Int'l Conf. Image Process. (ICIP 2011)*, Sep. 2011, pp. 1089–1092.

[45] M. Solh and G. AlRegib, "Hierarchical hole-filling for depth-based view synthesis in FTV and 3D video," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 5, pp. 495–504, Sep. 2012.

[46] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Müller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-D video," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 453–465, Jun. 2011.

[47] L. Tran, R. Khoshabeh, A. Jain, C. Pal, and T. Nguyen, "Spatially consistent view synthesis with coordinate alignment," in *2011 IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP 2011)*, May 2011, pp. 905–908.

[48] Y. Chen, W. Wan, M. Hannuksela, J. Zhang, H. Li, and M. Gabbouj, "Depth-level-adaptive view synthesis for 3D video," in *2010 IEEE Int'l Conf. Multimedia Expo (ICME 2010)*, Jul. 2010, pp. 1724–1729.

[49] D. Tian, A. Vetro, and M. Brand, "A trellis-based approach for robust view synthesis," in *2011 IEEE Int'l Conf. Image Process. (ICIP 2011)*, Sep. 2011, pp. 605–608.

[50] L. Do, S. Zinger, and P. H. N. de With, "Quality improving techniques for free-viewpoint DIBR," in *2010 SPIE Stereosc. Display Appl. XXI*, 2010, pp. 75 240I/1–10.

[51] P.-K. Tsung, P.-C. Lin, L.-F. Ding, S.-Y. Chien, and L.-G. Chen, "Single iteration view interpolation for multiview video applications," in *2009 IEEE 3DTV Conf. (3DTV-CON 2009)*, May 2009, pp. 1–4.

[52] J. G. M. Gonçalves and V. Sequeira, "Sensor-based Depth Capturing," in *3D Videocommunication*. Wiley & Sons, Ltd, 2005, ch. 16, pp. 299–313.

[53] R. Lange, "3D Time-of-Flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology," Ph.D. dissertation, University of Siegen, 2000.

[54] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, "CW Matricial Time-of-Flight Range Cameras," in *Time-of-Flight Cameras and Microsoft Kinect^{TM}*. Springer, 2012, ch. 2, pp. 17–32.

[55] F. Remondino and D. Stoppa, Eds., *TOF Range-Imaging Cameras*. Springer Heidelberg, 2013.

[56] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, "Microsoft Kinect$^{\text{TM}}$ Range Camera," in *Time-of-Flight Cameras and Microsoft Kinect^{TM}*. Springer, 2012, ch. 3, pp. 33–47.

[57] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *2011 IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR 2011)*, Jun. 2011, pp. 1297–1304.

[58] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *ACM Symp. User Interface Software Techn.*, 2011, pp. 559–568.

[59] L. Cruz, D. Lucio, and L. Velho, "Kinect and rgbd images: Challenges and applications," in *2012 SIBGRAPI Conf. Graphics, Patterns and Images Tutorials (SIBGRAPI-T 2012)*, Aug. 2012, pp. 36–49.

[60] S. Mehrotra, Z. Zhang, Q. Cai, C. Zhang, and P. Chou, "Low-complexity, near-lossless coding of depth maps from kinect-like depth cameras," in *2011 IEEE Int'l Work. Multimedia Signal Process. (MMSP 2011)*, Oct. 2011, pp. 1–6.

[61] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int'l J. Comput. Vis.*

[62] L. Nalpantidis, G. C. Sirakoulis, and A. Gasteratos, "Review of stereo vision algorithms: From software to hardware," *Int'l J. Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.

[63] M. Tanimoto, T. Fujii, M.P. Tehrani, M. Wildeboer, Depth Estimation Reference Software (DERS) 4.0, ISO/IEC JTC 1/SC 29/WG 11 Doc. N16605, London (UK), Jun. 2009.

[64] G. Sourimant, "A simple and efficient way to compute depth maps for multi-view videos," in *2010 IEEE 3DTV Conf. (3DTV-CON 2010)*, Jun. 2010, pp. 1–4.

[65] G. J. Sullivan, "Standards-based approaches to 3D and multiview video coding," in *2009 SPIE Appl. Digit. Image Process. XXXII*, 2009, pp. 74 430Q/1–8.

[66] Y. Chen, Y.-K. Wang, K. Ugur, M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. Advances Signal Process.*, vol. 2009, 2009.

[67] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P. de With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Process.: Image Commun.*, vol. 24, no. 1-2, pp. 73–88, Jan. 2009.

[68] E. Bosc, V. Jantet, M. Pressigout, L. Morin, and C. Guillemot, "Bit-rate allocation for multi-view video plus depth," in *2011 IEEE 3DTV Conf. (3DTV-CON 2011)*, May 2011, pp. 1–4.

[69] E. Bosc, P. Riou, M. Pressigout, and L. Morin, "Bit-rate allocation between texture and depth: Influence of data sequence characteristics," in *2012 IEEE 3DTV Conf. (3DTV-CON 2012)*, Oct. 2012, pp. 1–4.

[70] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao, "Joint video/depth rate allocation for 3D video coding based on view synthesis distortion model," *Signal Process.: Image Commun.*, vol. 24, no. 8, pp. 666–681, Sep. 2009.

[71] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Motion vector sharing and bitrate allocation for 3D video-plus-depth coding," *EURASIP J. Appl. Signal Process.*, vol. 2009, pp. 1–13, Jan. 2008.

[72] ISO/IEC JTC1/SC29/WG11, Call for Proposals on 3D Video Coding Technology , Doc. N12036, Geneva (CH), Mar. 2011.

[73] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[74] ISO/IEC JTC 1/SC 29/WG 11, Text of ISO/IEC DIS 23008-2 High Efficiency Video Coding, Doc. N12935, Stockholm, Sweden, Jul. 2012.

[75] ISO/IEC JTC1/SC29/WG11, Overview of 3DV coding tools proposed in the CfP, Doc. N12348, Geneva (CH), Dec. 2011.

[76] J. Lee, H. Wey, and D.-S. Park, "A novel approach for efficient multi-view depth map coding," in *2010 Picture Coding Symp. (PCS 2010)*, Dec. 2010, pp. 302–305.

[77] B. Oh, H.-C. Wey, and D.-S. Park, "Plane segmentation based intra prediction for depth map coding," in *2012 Picture Coding Symp. (PCS 2012)*, May 2012, pp. 41–44.

[78] K.-J. Oh, S. Yea, A. Vetro, and Y.-S. Ho, "Depth reconstruction filter and down/up sampling for depth coding in 3-D video," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 747–750, Sep. 2009.

[79] ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 JCT-3V, JCT-3V AHG Report: 3D Coding Tool Testing (AHG6), Doc. JCT3V-C0006, Geneva (CH), Jan. 2013.

[80] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," in *2009 IEEE Int'l Conf. Image Process. (ICIP 2009)*, Nov. 2009, pp. 721–724.

[81] ——, "Depth map coding with distortion estimation of rendered view," in *2010 SPIE Visual Inf. Process. Commun.*, Jan. 2010, pp. 75 430B/1–10.

[82] Q. Zhang, P. An, Y. Zhang, and Z. Zhang, "Efficient rendering distortion estimation for depth map compression," in *2011 IEEE Int'l Conf. Image Process. (ICIP 2011)*, Sep. 2011, pp. 1105–1108.

[83] Y. Morvan, P. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," in *2006 SPIE Stereosc. Display and Appl.*, vol. 6055, 2006.

[84] Y. Morvan, D. Farin, and P. de With, "Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images," in *2007 IEEE Int'l Conf. Image Process. (ICIP 2007)*, vol. 5, Oct. 2007, pp. 105–108.

[85] L. Lucas, N. Rodrigues, C. Pagliari, E. da Silva, and S. de Faria, "Efficient depth map coding using linear residue approximation and a flexible prediction framework," in *2012 IEEE Int'l Conf. Image Process. (ICIP 2012)*, Sep. 2012, pp. 1305–1308.

[86] G. Shen, W.-S. Kim, S. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *2010 Picture Coding Symp. (PCS 2010)*, Dec. 2010, pp. 566–569.

[87] W.-S. Kim, S. Narang, and A. Ortega, "Graph based transforms for depth video coding," in *2012 IEEE Int'l Conf. Acoust. Speech Signal Process. (ICASSP 2012)*, Mar. 2012, pp. 813–816.

[88] G. Cheung, A. Kubota, and A. Ortega, "Sparse representation of depth maps for efficient transform coding," in *2010 Picture Coding Symp. (PCS 2010)*, Dec. 2010, pp. 298–301.

[89] Y. Zhao, C. Zhu, Z. Chen, and L. Yu, "Depth No-Synthesis-Error Model for View Synthesis in 3-D Video," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2221–2228, Aug. 2011.

[90] G. Cheung, J. Ishida, A. Kubota, and A. Ortega, "Transform domain sparsification of depth maps using iterative quadratic programming," in *2011 IEEE Int'l Conf. Image Process. (ICIP 2011)*, Sep. 2011, pp. 129–132.

[91] G. Cheung, W.-S. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth map coding using graph based transform and transform domain sparsification," in *2011 IEEE Int'l Work. Multimedia Signal Process. (MMSP 2011)*, Oct. 2011, pp. 1–6.

[92] D. Graziosi, N. Rodrigues, C. Pagliari, E. da Silva, S. de Faria, M. Perez, and M. de Carvalho, "Multiscale recurrent pattern matching approach for depth map coding," in *2010 Picture Coding Symp. (PCS 2010)*, Dec. 2010, pp. 294–297.

[93] S. Shimizu, H. Kimata, S. Sugimoto, and N. Matsuura, "Block-adaptive palette-based prediction for depth map coding," in *2011 IEEE Int'l Conf. Image Process. (ICIP 2011)*, Sep. 2011, pp. 117–120.

[94] P. Zanuttigh and G. Cortelazzo, "Compression of depth information for 3d rendering," in *2009 IEEE 3DTV Conf. (3DTV-CON 2009)*, May 2009, pp. 1–4.

[95] Y. Li and L. Sun, "A novel upsampling scheme for depth map compression in 3DTV system," in *2010 Picture Coding Symp. (PCS 2010)*, Dec. 2010, pp. 186–189.

[96] S. Liu, P. Lai, D. Tian, C. Gomila, and C. Chen, "Joint trilateral filtering for depth map compression," in *2010 SPIE Visual Commun. Image Process.*, 2010, pp. 77 440F/1–10.

[97] I. Daribo, H. Saito, R. Furukawa, S. Hiura, and N. Asada, "Effects of Wavelet-Based Depth Video Compression," in *3D-TV System with Depth-Image-Based Rendering*, C. Zhu, Y. Zhao, L. Yu, and M. Tanimoto, Eds. Springer, 2013, ch. 10, pp. 277–298.

[98] A. Sanchez, G. Shen, and A. Ortega, "Edge-preserving depth-map coding using graph-based wavelets," in *2009 Asilomar Conf. Signals, Systems and Comput.*, Nov. 2009, pp. 578–582.

[99] R. Mathew, P. Zanuttigh, and D. Taubman, "Highly scalable coding of depth maps with arc breakpoints," in *2012 Data Compression Conf. (DCC 2012)*, Apr. 2012, pp. 42–51.

[100] R. Mathew, D. Taubman, and P. Zanuttigh, "Scalable depth maps with r-d optimized embedding," in *2012 IEEE Int'l Work. Multimedia Signal Process. (MMSP 2012)*, Sep. 2012, pp. 266–271.

[101] T.-Y. Chung, S. Sull, and C.-S. Kim, "Frame loss concealment for stereoscopic video plus depth sequences," *IEEE Trans. Consum. Electron.*, vol. 57, no. 3, pp. 1336–1344, Aug. 2011.

[102] A. Tekalp and M. Civanlar, "Efficient Transport of 3DTV," in *Three-Dimensional Television.* Springer Heidelberg, 2008, ch. 10, pp. 351–369.

[103] A. Norkin, M. O. Bici, A. Aksay, C. Bilen, A. Gotchev, G. Akar, K. Egiazarian, and J. Astola, "Multiple Description Coding and its Relevance to 3DTV," in *Three-Dimensional Television.* Springer Heidelberg, 2008, ch. 11, pp. 371–426.

[104] K. Kim, G. Park, and D. Suh, "Bit-plane-based lossless depth-map coding," *Opt. Eng.*, vol. 49, no. 6, Jun. 2010.

[105] L. Cappellari, C. Cruz-Reyes, G. Calvagno, and J. Kari, "Lossy to lossless spatially scalable depth map coding with cellular automata," in *2009 Data Compression Conf. (DCC 2009)*, ser. DCC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 332–341.

[106] I. Şchiopu and I. Tăbuş, "MDL segmentation and lossless compression of depth images," in *4th Work. on Inf. Th. Meth. in Sc. and Eng. (WITMSE 2011)*, Helsinki, Finland, Aug. 2011.

[107] J. Heo and Y.-S. Ho, "Improved context-based adaptive vinary arithmetic coding over H.264/AVC for lossless depth map coding," *IEEE Signal Process. Lett.*, vol. 17, no. 10, pp. 835–838, Oct. 2010.

[108] L. Onural, "Current Research Trends," in *3D Video Technologies*. SPIE Press, 2011, ch. 4, pp. 61–76.

[109] [Online]. Available: http://www.3diim-cluster.eu

[110] [Online]. Available: http://www.3dtv-research.org

[111] [Online]. Available: http://muted.zcu.cz

[112] [Online]. Available: http://www.cse.dmu.ac.uk/~heliumusr

[113] [Online]. Available: http://sp.cs.tut.fi/mobile3dtv

[114] [Online]. Available: https://sites.google.com/site/tolgacapin/research-projects

[115] [Online]. Available: http://www.hitech-projects.com/euprojects/3d4you/www.3d4you.eu/index-2.html

[116] [Online]. Available: http://www.3dpresence.eu

[117] [Online]. Available: http://www.20203dmedia.eu

[118] [Online]. Available: http://www.diomedes-project.eu

[119] [Online]. Available: http://www.muscade.eu

[120] [Online]. Available: http://ict-skymedia.eu/skymedia

[121] [Online]. Available: http://www.fascinate-project.eu

[122] [Online]. Available: http://www.ict-romeo.eu

[123] [Online]. Available: http://3d-scene.eu

[124] [Online]. Available: http://www.reveriefp7.eu

[125] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan.

[126] F. Pereira, "Distributed Video Coding: basics, main solutions and trends," in *2009 IEEE Int'l Conf. Multimedia Expo (ICME 2009)*, Jul. 2009, pp. 1592–1595.

[127] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[128] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[129] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed Monoview and Multiview Video Coding," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 67–76, Sep. 2007.

[130] G. Petrazzuoli, M. Cagnazzo, F. Dufaux, and B. Pesquet-Popescu, "Wyner-Ziv coding for depth maps in multiview video-plus-depth," in *2011 IEEE Int'l Conf. Image Process. (ICIP 2011)*, Sep. 2011., pp. 1817–1820.

[131] F. Dufaux, W. Gao, S. Tubaro, and A. Vetro, "Distributed Video Coding: trends and perspectives," *J. Image Video Process.*, vol. 2009, pp. 1–13, Feb. 2009.

[132] M. Zamarin, S. Milani, P. Zanuttigh, and G. M. Cortelazzo, "A novel multi-view image coding scheme based on view-warping and 3D-DCT," *J. Visual Commun. Image Represent.*, vol. 21, no. 5-6, pp. 462–473, 2010.

[133] Y. Niu, X. Wu, G. Shi, and X. Wang, "Edge-based perceptual image coding," *IEEE Trans. Image Process*, vol. 21, no. 4, pp. 1899–1910, 2012.

[134] M. Mainberger, A. Bruhn, J. Weickert, and S. Forchhammer, "Edge-based compression of cartoon-like images with homogeneous diffusion," *Pattern Recogn.*, vol. 44, no. 9, pp. 1859–1873, 2011.

[135] "Repository vision.middlebury.edu: Stereo datasets," http://vision.middlebury.edu/stereo.

[136] X. Huang and S. Forchhammer, "Cross-band noise model refinement for transform domain Wyner-Ziv video coding," *Signal Process.: Image Commun.*, vol. 27, no. 1, pp. 16–30, 2012.

[137] R. Szeliski, *Computer Vision: algorithms and applications*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[138] Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Trans. Image Process*, vol. 11, no. 12, pp. 1442–1449, 2002.

[139] X. Huang, L. Rakêt, H. Van Luong, M. Nielsen, F. Lauze, and S. Forchhammer, "Multi-hypothesis transform domain Wyner-Ziv video coding including optical flow," in *2011 IEEE Int'l Work. Multimedia Signal Process. (MMSP 2011)*, 2011, pp. 1–6.

[140] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int'l J. Comput. Vis.*, vol. 59, pp. 167–181, Sep. 2004.

[141] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[142] JPEG-LS, "Lossless and near-lossless compression of continuous-tone still images," in *Int'l Standard ISO/IEC 14495*, 1999.

[143] D. Taubman and M. Marcellin, *JPEG2000: image compression fundamentals, standards, and practice*. Kluwer Academic Publishers, 2002.

[144] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[145] JBIG, "Progressive Bi-level Image Compression," in *Int'l Standard ISO/IEC 11544, ITU-T Rec. T.82*, 1993.

[146] G. Shen, W.-S. Kim, A. Ortega, J. Lee, and H. Wey, "Edge-aware intra prediction for depth-map coding," in *2010 IEEE Int'l Conf. Image Process. (ICIP 2010)*, Sep. 2010, pp. 3393–3396.

[147] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER codec: architecture, techniques and evaluation," in *2007 Picture Coding Symp. (PCS 2007)*, 2007.

[148] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD-Curves," in *ITU-T SG 16 Q.6, VCEG-M33, Austin, Texas, USA*, Apr. 2001.

[149] L. L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert, "Motion compensated frame interpolation with a symmetric optical flow constraint," in *Advances in Visual Computing*, G. Bebis *et al.*, Ed. Springer, 2012, vol. 7431 of *Lecture notes in computer science*, pp. 447–457.

[150] J. Gautier, O. Le Meur, and C. Guillemot, "Efficient depth map compression based on lossless edge coding and diffusion," in *2012 Picture Coding Symp. (PCS 2012)*, 2012, pp. 81–84.

[151] A. Vetro and D. Tian, "Analysis of 3D and multiview extensions of the emerging HEVC standard," in *2012 SPIE Appl. Digit. Image Process. XXXV*, 2012, pp. 84 990Y/1–7.